

O'REILLY®

# Google Cloud: Associate Cloud Engineer Bootcamp - Day 1





# Prereqs for Course

- No previous experience with Google Cloud
- Some exposure to working on the cloud recommended
- Basic understanding of deploying software on-premises



# Day 1: Course Schedule

- Resource Hierarchy on Google Cloud
- Infrastructure-as-a-Service on the Google Cloud
- Networking on the Google Cloud
- Connecting Networks
- Google App Engine
- Google Cloud Run Functions
- Google Cloud Run
- Instance Groups



# Day 2: Course Schedule

- Taxonomy of Storage Solutions
- Google Cloud Storage
- Containers and Kubernetes
- Load Balancing
- Identity and Access Management
- IAM Best Practices
- Organization Policy Service
- Billing Accounts
- Batch Processing vs. Stream Processing
- Pub/Sub



# Prereqs for Hands-on Demos

- Create a free Google Cloud account
- <https://console.cloud.google.com/>
- Enable billing on that account
- Please watch the getting set up video linked here:
- <https://drive.google.com/drive/folders/130rcJUmsy4LANX-7iWasu7KmuvFULkSf?usp=sharing>



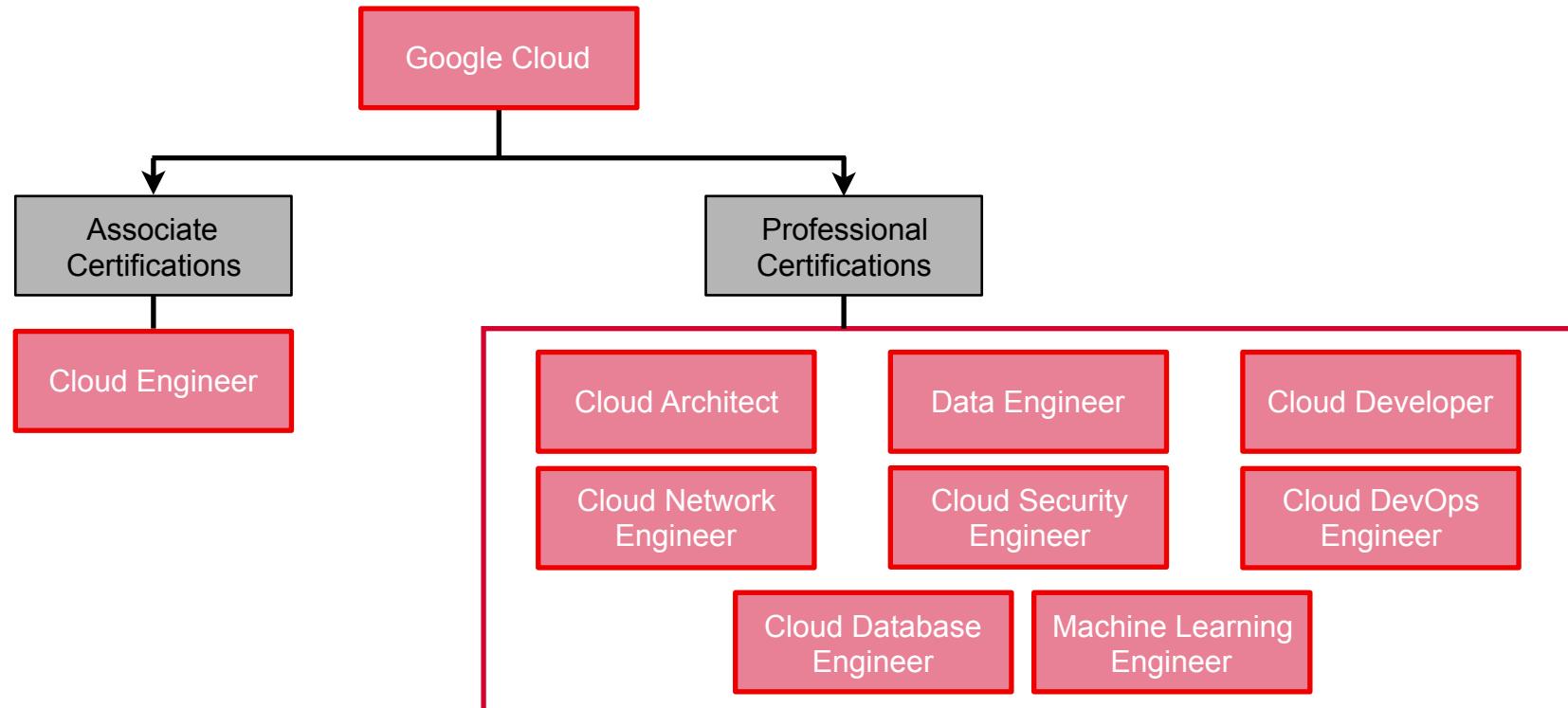
# Associate Cloud Engineer

- Test duration: 2 hours
- Registration fee: \$125 + taxes
- Languages: English, Japanese, Spanish, Portuguese
- Exam format: 50-60 multiple choice and multiple select questions
- Recommended: 6+ months hands-on experience with the Google Cloud



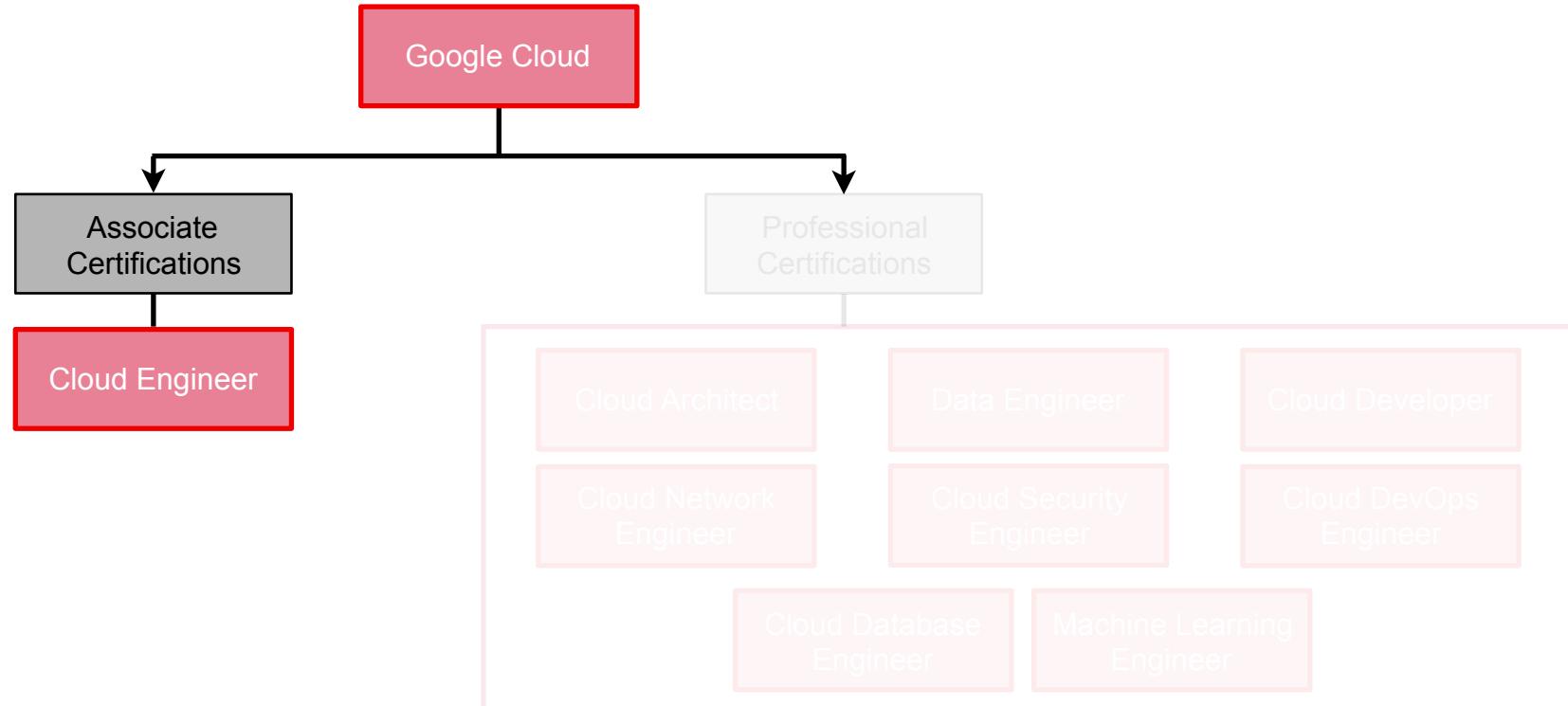


# Google Cloud Certifications





# Google Cloud Certifications

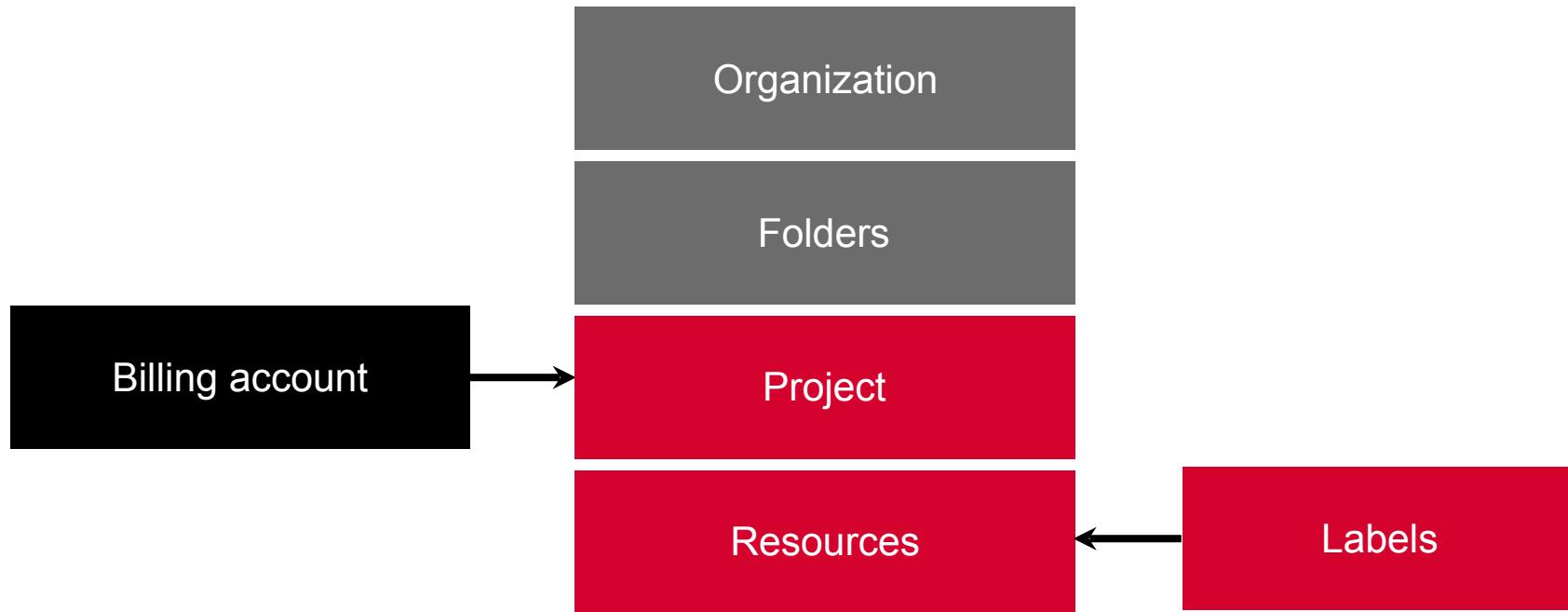


# Resource Hierarchy on Google Cloud



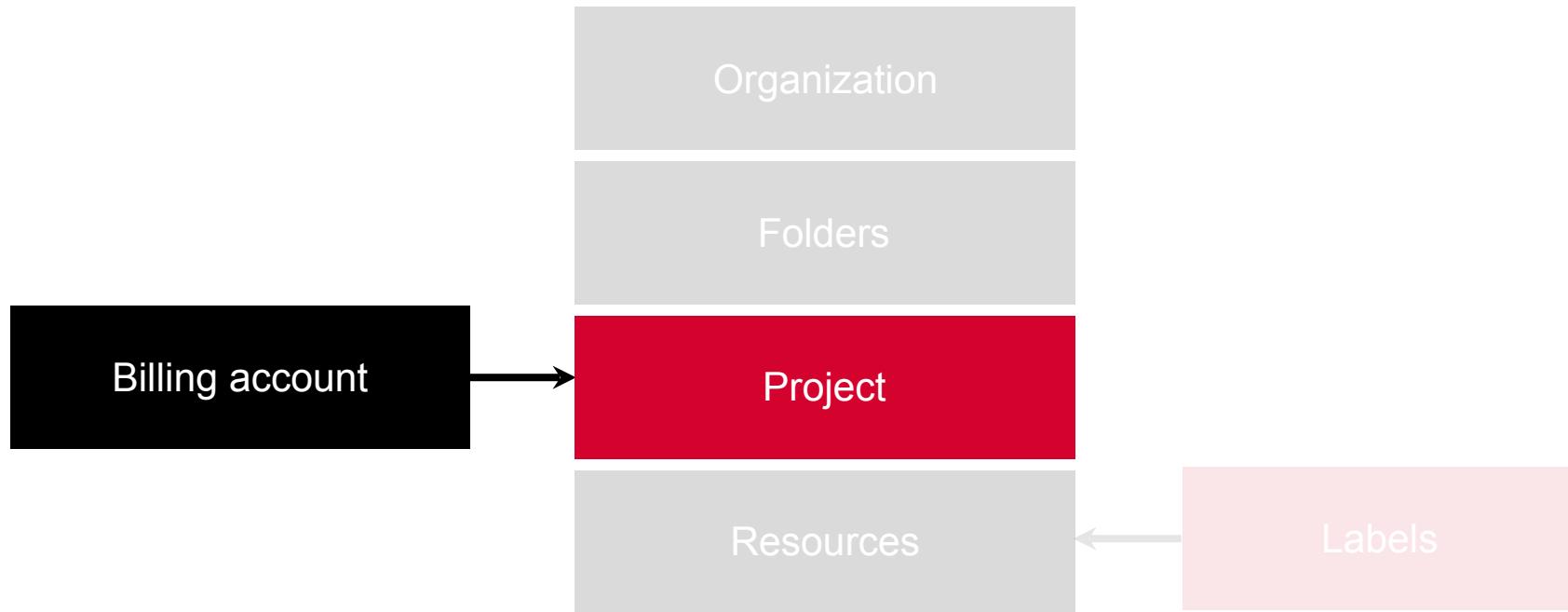


# Resource Hierarchy of Components



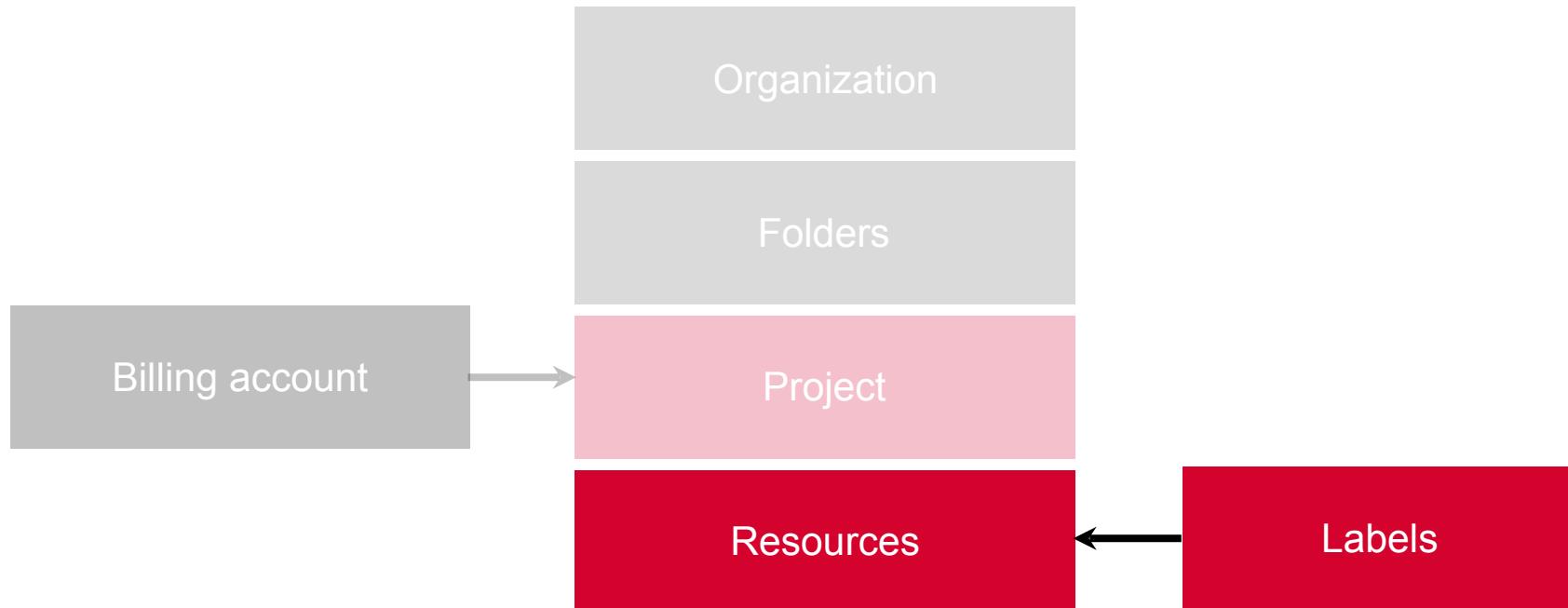


# Billing Accounts Are Associated with Projects





# Labels Are Applied to Resources





# Organization

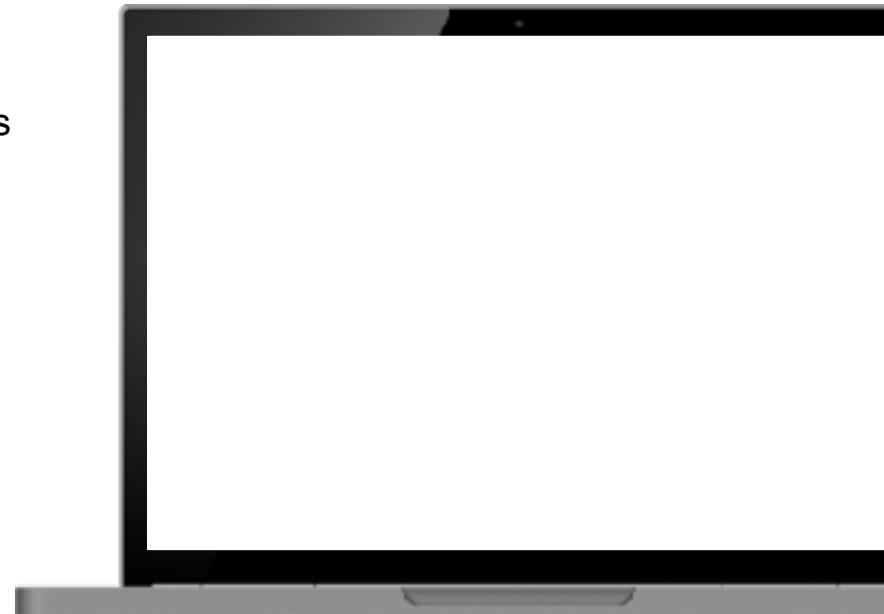
- Top of resource hierarchy
- Contains projects and folders
- Identities come from G Suite or a Cloud Identity account
- IAM policies are inherited down into projects and resources
- Central control for all resources
- Projects belong to the organization, not employees
- Can grant organization level roles





# Folders

- Grouping mechanism within an organization
- Logical group of projects
- Can set IAM policies to administer multiple projects
- Model legal entities, departments, and teams





# Projects

- Container for billable resources
- Some resources can be used for free
- For all others, billing account needs to be linked
- Required resource for using Google Cloud services





# Resources

- Any component that incurs billing
- Must exist within project
- Can set resource-level IAM
- Additional IAM policies inherited from organization, folder, project
- Lowest level of the hierarchy





# Using Google Cloud Resources

**Cloud Console**

**Cloud Shell**

**Command-line Tools**

**APIs and Client Libraries**

# Infra-as-a- Service on the Google Cloud



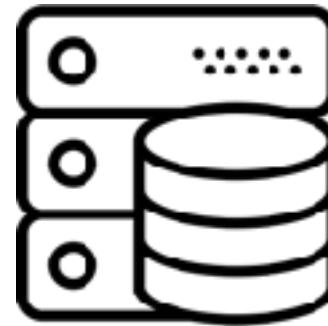


# Choices in Computing



**Compute**

Where is code executed and how?



**Storage**

Where is data stored?

Networking, logging, are choices made after this fundamental decision

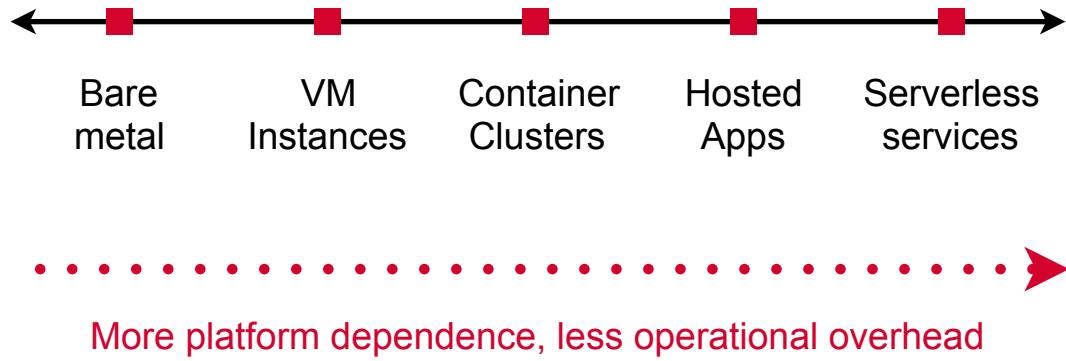


# Compute Choices



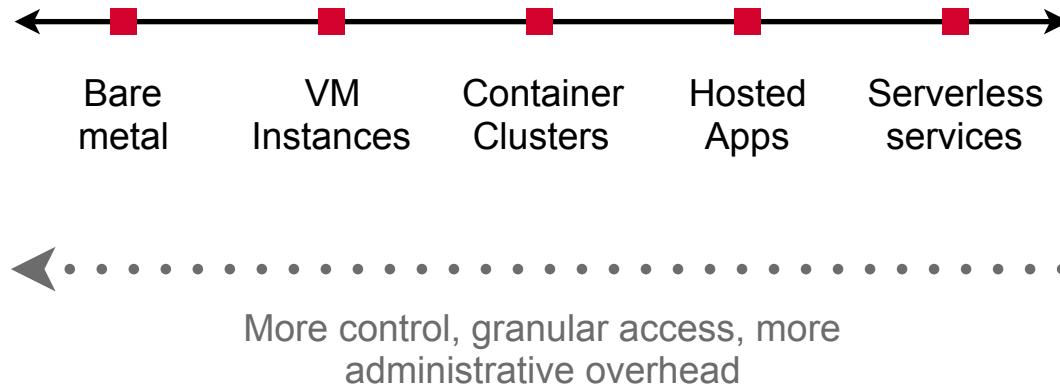


# Compute Choices





# Compute Choices



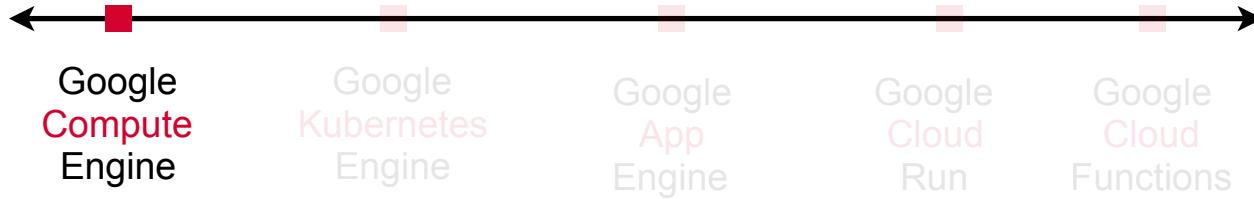


# Google Cloud Compute Choices





# Google Cloud Compute Choices





# IaaS vs. Bare Metal

## Bare Metal

- Apps run on OS which runs on hardware
- Less portable
- CPUs
- Full burden of ops and admin

## IaaS

- Hypervisor between apps and hardware
- More portable
- vCPUs
- Much of ops burden managed by service provider



# Google Cloud Internals



**Zone**

Availability zone  
(similar to a  
datacenter)



**Region**

Set of zones with high-  
speed network links



**Network**

User-controlled IP  
addresses, subnets and  
firewalls



# Google Cloud Internals



**Zone**

“us-central1-a”



**Region**

“us-central1”



**Network**

“default”



# Google Cloud Internals



**Zone**

“asia-south1-a”



**Region**

“asia-south1”



**Network**

“default”



**Resources provisioned in zones are NOT  
fault tolerant - regional resources can  
survive zone downtimes**



# Configuration Choices

## Machine Family

General purpose, compute optimized, memory optimized, accelerator-optimized

## Machine Series

Machines have generation numbers where higher generations have newer features

## Machine Type

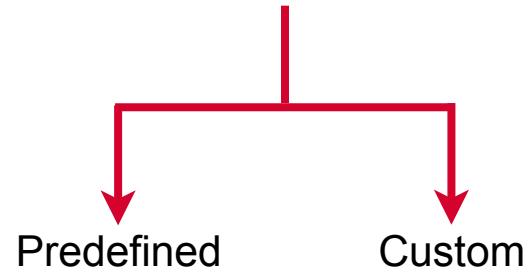
vCPUs count, memory capacity, and storage capacity

## Base Image

Public (free or premium), custom, snapshots from boot disks

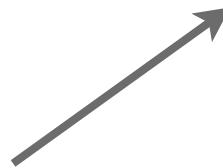
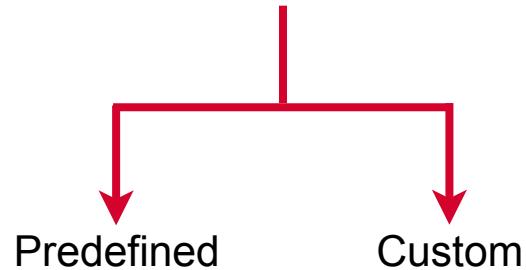


# Machine Type





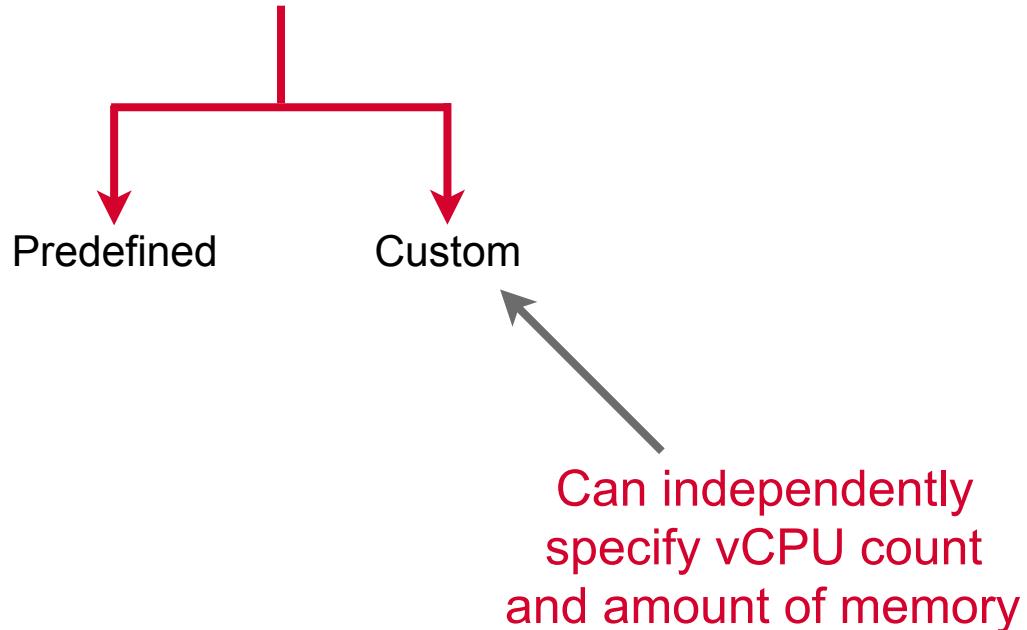
# Machine Type



Fixed set of types with  
fixed ratios of memory  
to vCPU count

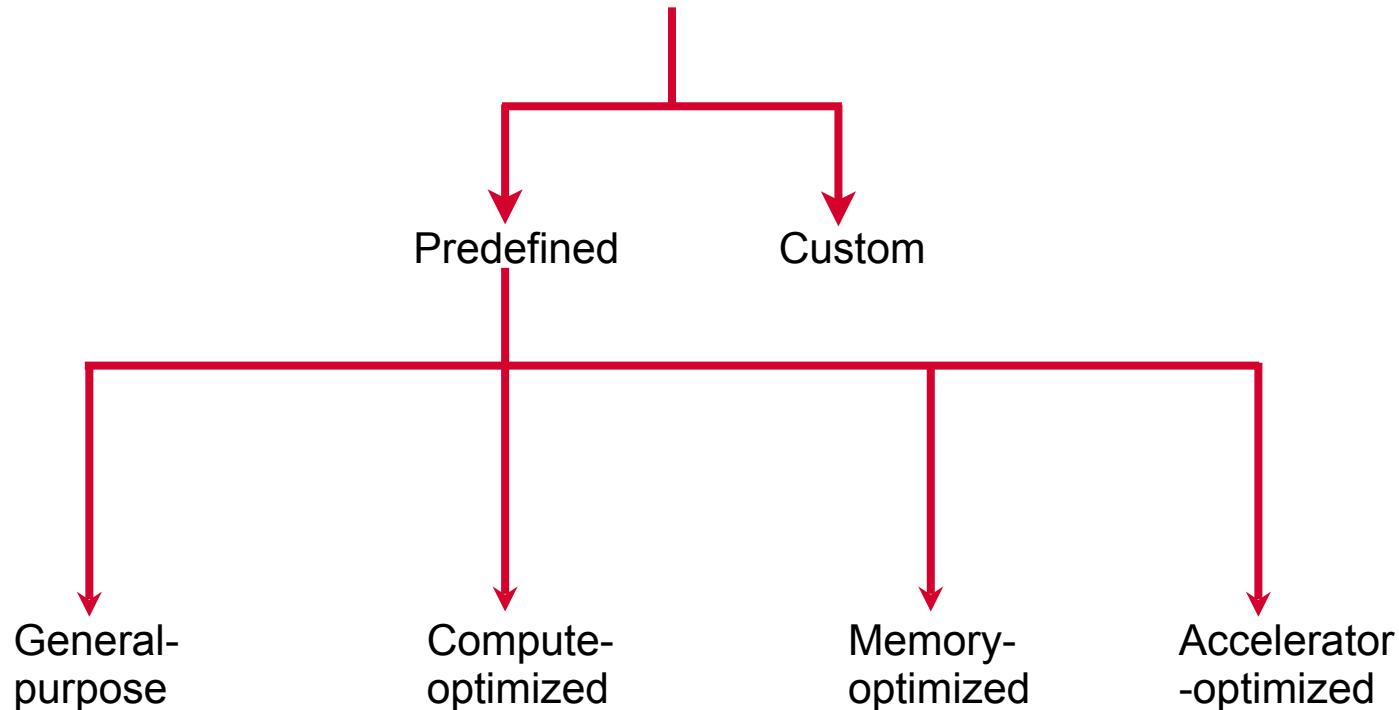


# Machine Type





# Machine Type





# General Purpose Machines

- Day to day computing for known workloads
- **Best price-performance ratio**
- N1 first generation: 6.5GB of memory per vCPU
- N2 second generation: 8GB of memory per vCPU
  - More heavy duty workloads such as web serving, databases, applications use N2
- Can customize machine types
- Come in high-memory and high-cpu variants





# Compute-optimized Machines

- Compute intensive workloads
- Offer the **highest performance per core**
- C2 machine types
- Gaming, single-threaded applications,  
electronic design automation
- Custom machine types not supported





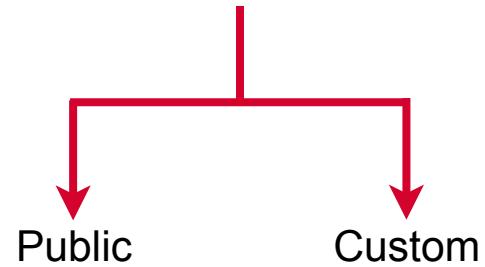
# Memory-optimized Machines

- Memory-intensive workloads
- Offer the **highest memory per core**
- Custom machine types not supported



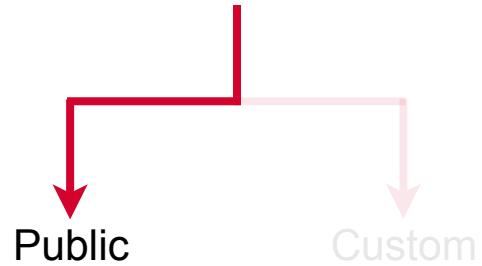


# Base Images





# Base Images

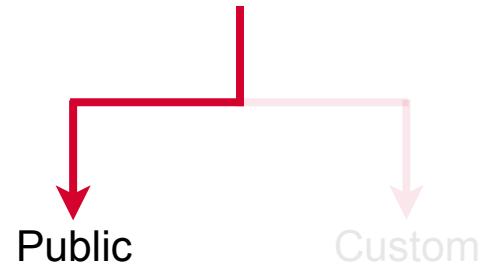


Provided and maintained by Google, open-source communities, and third-party vendors

All projects have access to these images and can use them to create instances



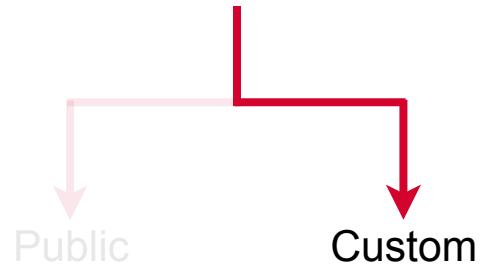
# Base Images



Linux, Windows, Container-optimized OS,  
SQL Server



# Base Images



Available only to your project

First, create a custom image from boot disks  
and other images; then, use the custom  
image to create an instance



# Spot VM Instances

An instance that you can create and run at a much lower price than normal instances. However, **GCE might terminate (preempt)** these instances if it requires access to those resources for other tasks.

May not always be available

Not covered by SLAs





# Preemptible VM Instances

Similar to Spot VMs (older product and will have fewer features than Spot VMs)

**Will definitely be preempted every 24 hours**

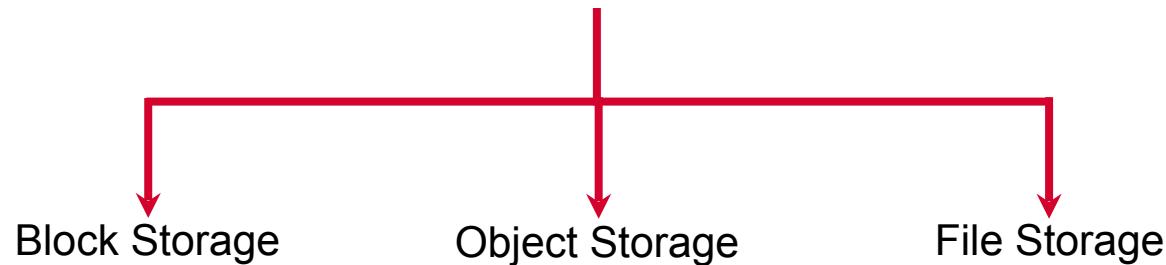
May not always be available

Not covered by SLAs



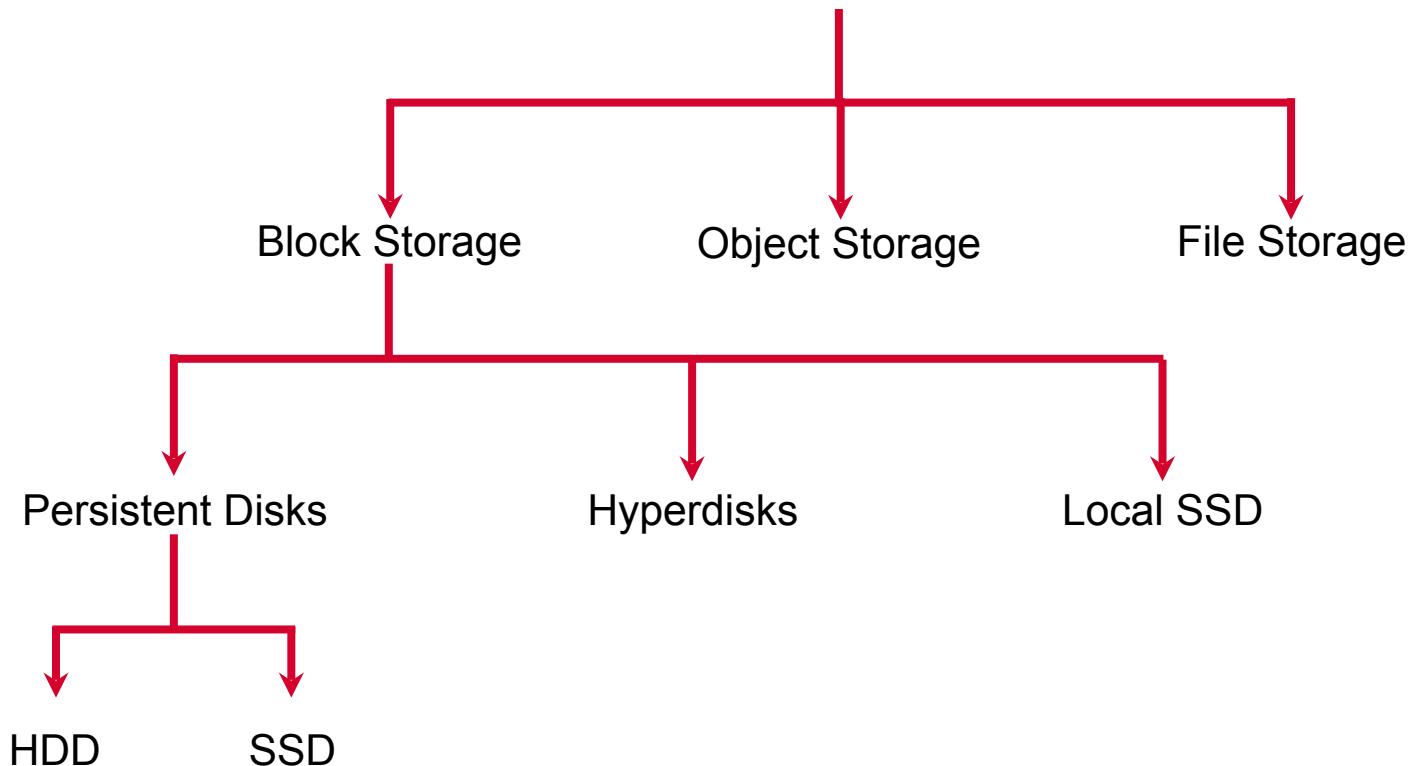


# Accessing Storage from VMs





# Accessing Storage from VMs





# Persistent Disks vs. Local SSDs

## Persistent Disks

- Network-attached storage
- Data redundancy built-in
- Bootable
- Durable
- HDD or SSD
- 64TB max for one volume
- Create snapshots or images
- Relatively slow

## Local SSDs

- **Physically attached to instance**
- No data redundancy built-in
- Not bootable
- Not durable
- SSD for better performance
- 9TB max
- Cannot create snapshots or images
- **Very fast, especially for random access**



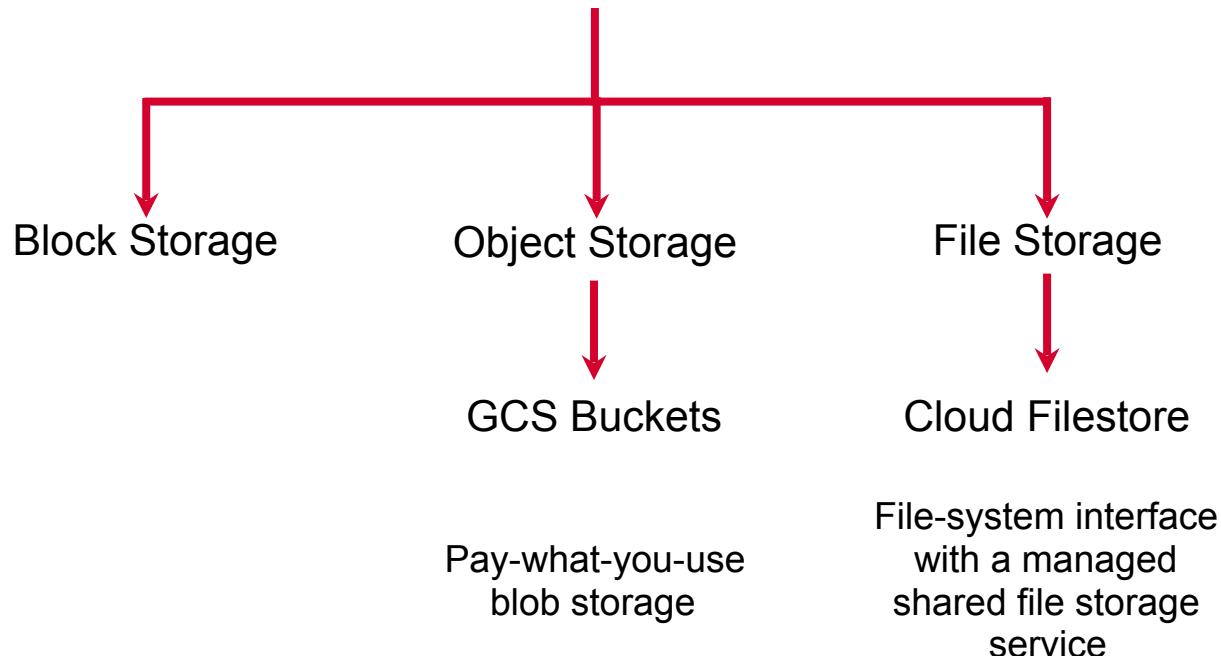
# Zonal or Regional Persistent Disks

Zonal persistent disks are associated with a single zone  
- not high availability

Regional persistent disks are associated with a region - high availability regional fault tolerance



# Accessing Storage from VMs





# **Choose Hyperdisks for very high performance for disk I/O**



# Persistent Disks vs. Buckets

## Persistent Disks

- Block storage
- Max 64TB in size
- **Pay what you allocate**
- Tied to GCE VMs
- Zonal (or regional) access

## Buckets

- Object storage
- Infinitely scalable
- Pay what you use
- Independent of GCE VMs
- Global access

O'REILLY®

# Snapshots and Images





# Image

- Binary file used to instantiate VM root disk
- Usually based off OS image
- Also contains boot loader
- Can also contain customizations
- Managed by GCP **image** service





# Snapshot

- Binary file with exact contents of persistent disk
- “Point-in-time” snapshot
- Managed by GCP **snapshot** service
- **Incremental backups possible too**
- Used to back up data from persistent disks





# Snapshots and Images

Conceptually very similar but many differences in nitty-gritty

# Compute Engine

Your team is running a batch processing job on Google Cloud that requires a large number of virtual machines (VMs). However, the current cost of using standard VMs is too high. You suspect that the batch workloads may be fault tolerant, but you need to confirm this through testing. What can you do to reduce costs while ensuring the job can tolerate potential interruptions?

- A. Migrate the batch processing job to use spot VMs, which are cheaper but can be interrupted, and run a simulation to verify if the workloads can recover from interruptions.
- B. Use sole-tenant VMs to dedicate hardware resources, ensuring high reliability but at a higher cost.
- C. Switch to higher-performance VMs to complete the job faster and reduce overall runtime costs without testing for fault tolerance.
- D. Implement autoscaling to dynamically adjust the number of standard VMs based on job load, but without testing for fault tolerance or using preemptible VMs.



# Compute Engine

Your team is running a batch processing job on Google Cloud that requires a large number of virtual machines (VMs). However, the current cost of using standard VMs is too high. You suspect that the batch workloads may be fault tolerant, but you need to confirm this through testing. What can you do to reduce costs while ensuring the job can tolerate potential interruptions?

- A. Migrate the batch processing job to use spot VMs, which are cheaper but can be interrupted, and run a simulation to verify if the workloads can recover from interruptions.**
- B. Use sole-tenant VMs to dedicate hardware resources, ensuring high reliability but at a higher cost.
- C. Switch to higher-performance VMs to complete the job faster and reduce overall runtime costs without testing for fault tolerance.
- D. Implement autoscaling to dynamically adjust the number of standard VMs based on job load, but without testing for fault tolerance or using preemptible VMs.



# Compute Engine

You are moving a critical application from your on-premises data center to Google Cloud. To ensure high availability, you want the application's data to remain accessible in the event of a failure in one zone. What should you do to ensure data availability during a zonal outage?

- A. Use zonal persistent disks for storage and set up automatic backups. During an outage, restore the backup to a new disk in a different zone and attach it to a new VM.
- B. Use zonal persistent disks for storage and manually replicate the data to another zone for recovery if an outage occurs.
- C. Use regional persistent disks for storage, which synchronously replicate data across zones, ensuring immediate availability if one zone fails.
- D. Use regional persistent disks with regular snapshots. In case of a zonal failure, restore data from the latest snapshot to a new disk in another zone and attach it to a VM.



# Compute Engine

You are moving a critical application from your on-premises data center to Google Cloud. To ensure high availability, you want the application's data to remain accessible in the event of a failure in one zone. What should you do to ensure data availability during a zonal outage?

- A. Use zonal persistent disks for storage and set up automatic backups. During an outage, restore the backup to a new disk in a different zone and attach it to a new VM.
- B. Use zonal persistent disks for storage and manually replicate the data to another zone for recovery if an outage occurs.
- C. Use regional persistent disks for storage, which synchronously replicate data across zones, ensuring immediate availability if one zone fails.**
- D. Use regional persistent disks with regular snapshots. In case of a zonal failure, restore data from the latest snapshot to a new disk in another zone and attach it to a VM.



# Compute Engine

You have a batch workload currently running on a single machine in your on-premises environment. The job runs for approximately 35 hours and is **not fault-tolerant**, meaning it cannot recover from interruptions. You need to move this workload to Google Cloud with minimal effort and ideally no major code changes, ensuring the job runs reliably without failure.

What should you do?

- A. Migrate the batch job to Compute Engine VMs to ensure minimal code changes and reliable execution.
- B. Move the batch job to a Kubernetes cluster on GKE and configure it for autoscaling.
- C. Use Compute Engine Spot VMs to reduce costs, even though the job may be interrupted and need to restart.
- D. Re-architect the batch job to run on Cloud Run for automatic scaling and minimal management.



# Compute Engine

You have a batch workload currently running on a single machine in your on-premises environment. The job runs for approximately 35 hours and is **not fault-tolerant**, meaning it cannot recover from interruptions. You need to move this workload to Google Cloud with minimal effort and ideally no major code changes, ensuring the job runs reliably without failure.

What should you do?

- A. **Migrate the batch job to Compute Engine VMs to ensure minimal code changes and reliable execution.**
- B. Move the batch job to a Kubernetes cluster on GKE and configure it for autoscaling.
- C. Use Compute Engine Spot VMs to reduce costs, even though the job may be interrupted and need to restart.
- D. Re-architect the batch job to run on Cloud Run for automatic scaling and minimal management.



# Networking on the Google Cloud





# Google Virtual Private Cloud

A VPC network, or just network, is a global, private, isolated virtual network partition that provides managed network functionality on the Google Cloud

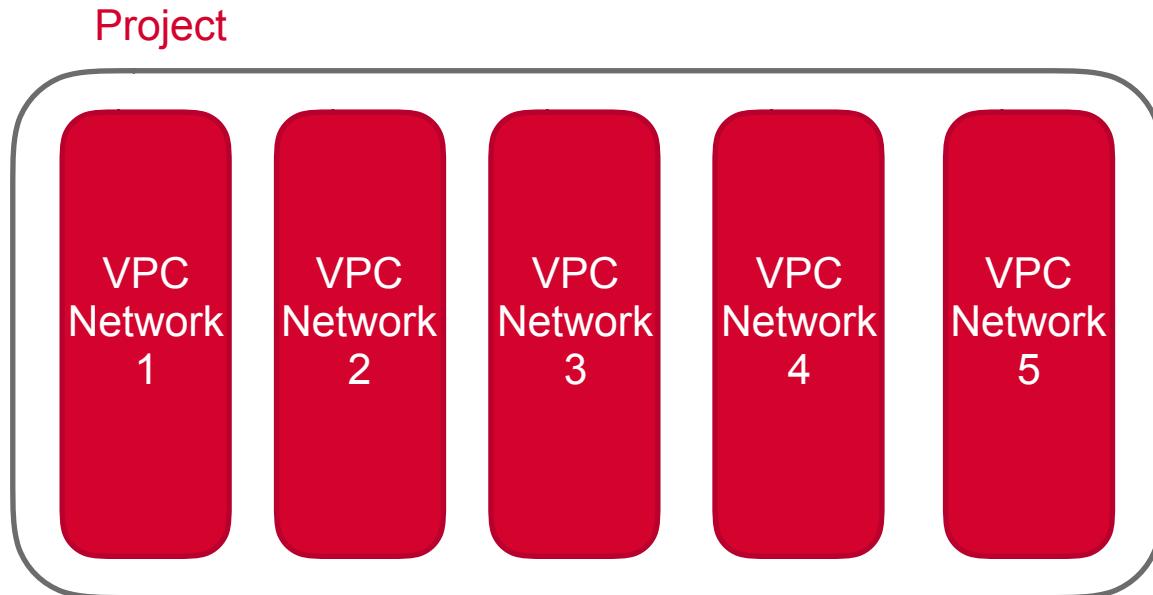


# Google Virtual Private Cloud

A VPC network, often just called a network, is a **global, private, isolated virtual network partition** that provides managed network functionality on Google Cloud



# Multiple VPCs in a Project





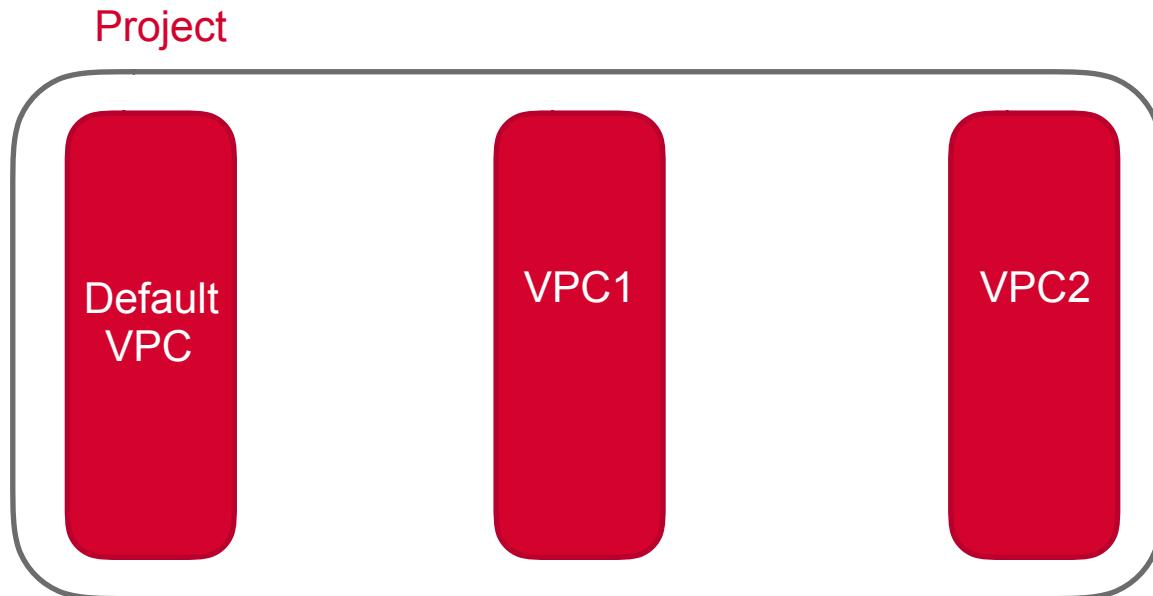
# Projects and VPCs

- VPCs are global resources on Google Cloud
- Each VPC must exist inside a project
- Default VPC **pre-created** in each project
- Can add additional VPCs
  - Auto Mode
  - Custom Mode



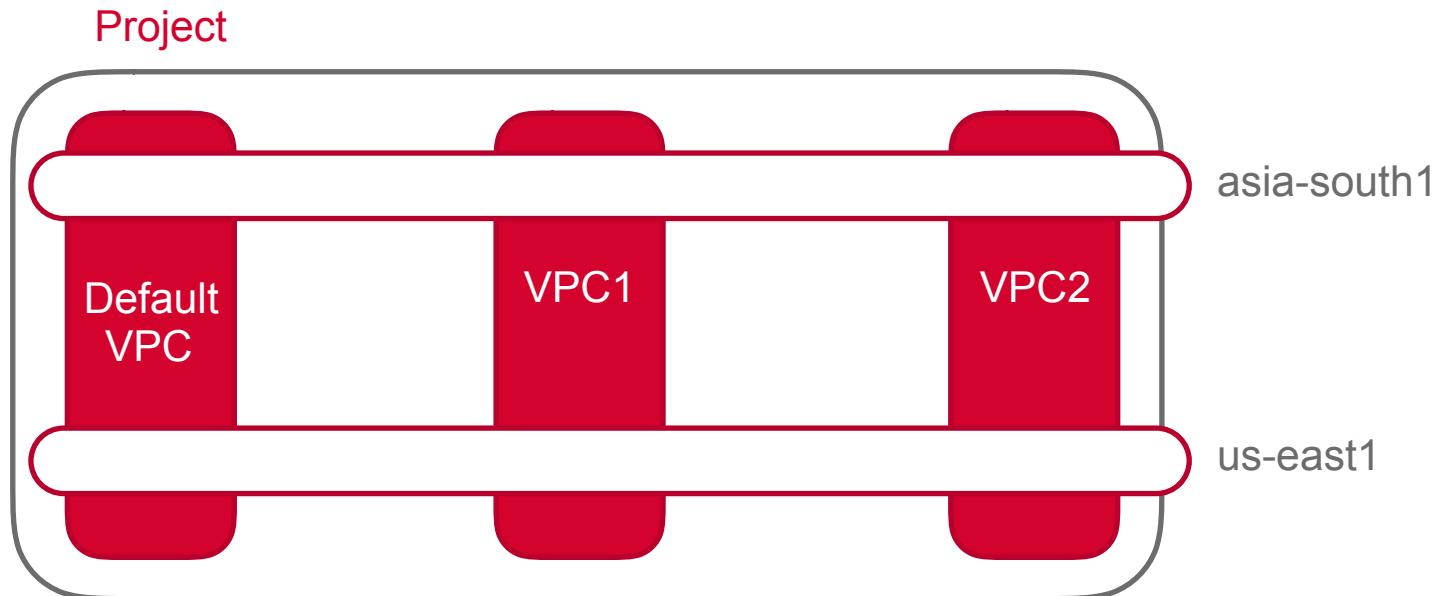


# VPCs Are Global



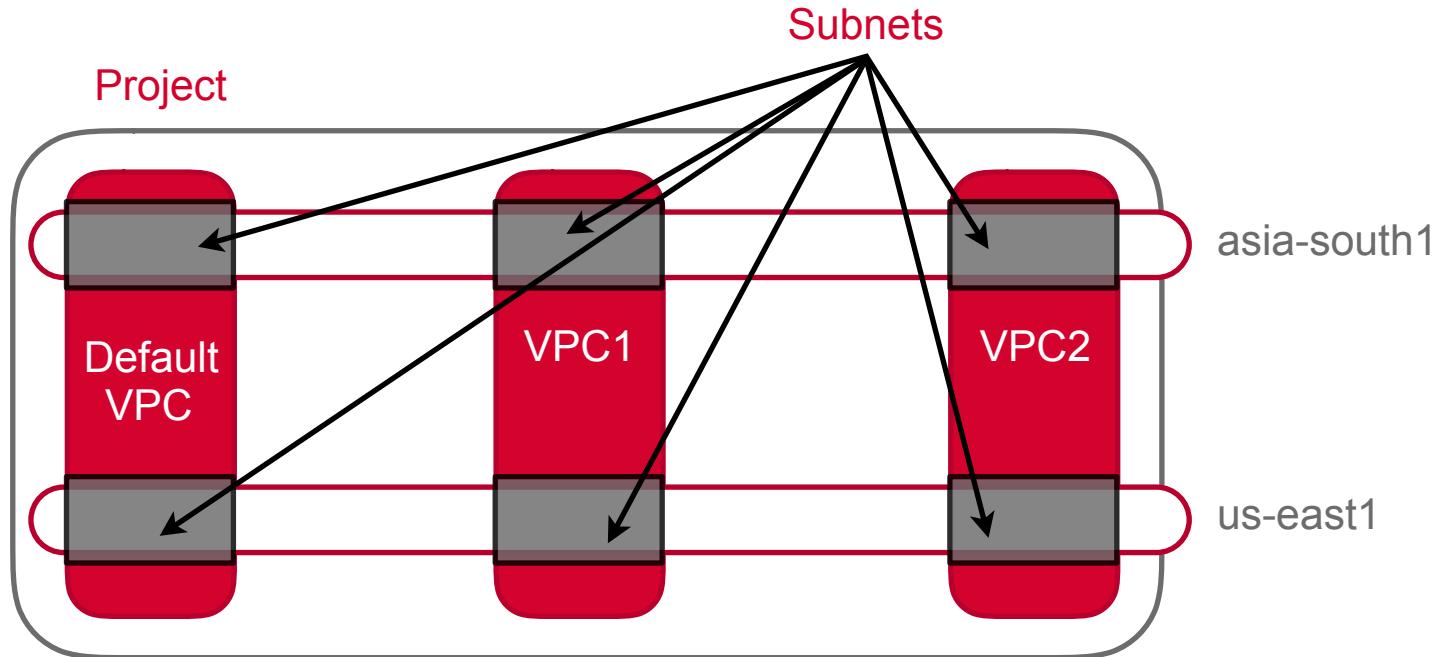


# VPCs Are Global



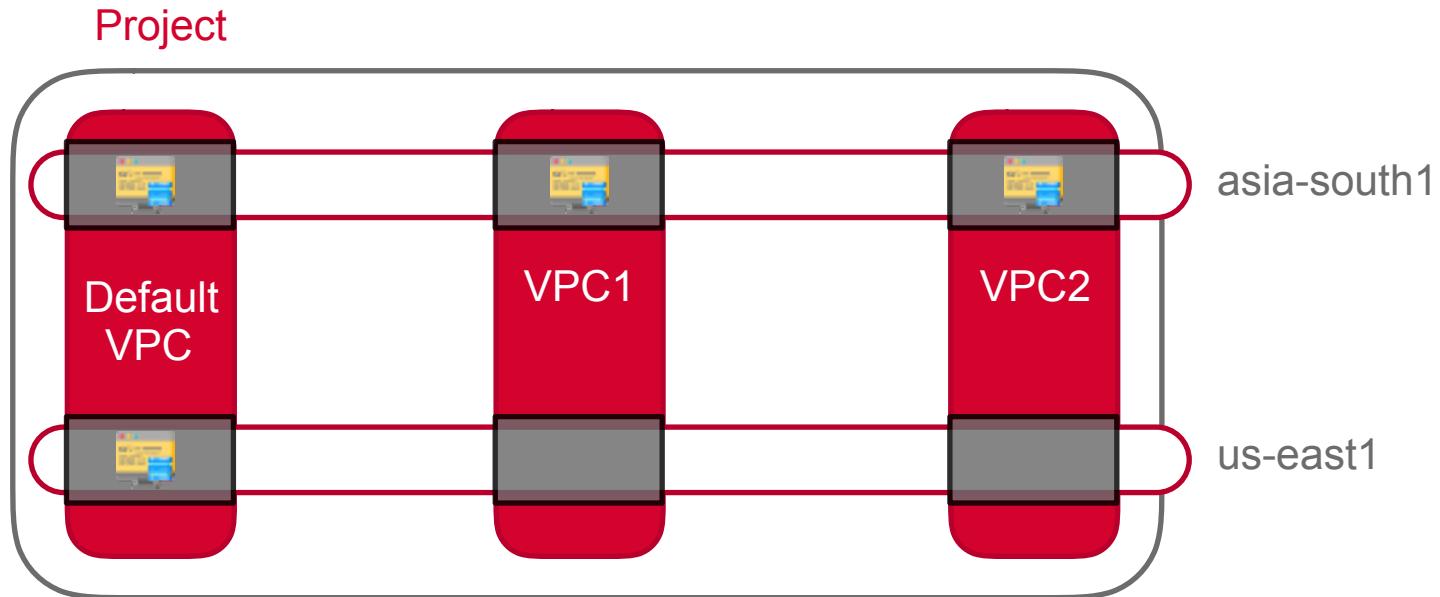


# Subnets in Each Region





# Resources Provisioned on Subnets





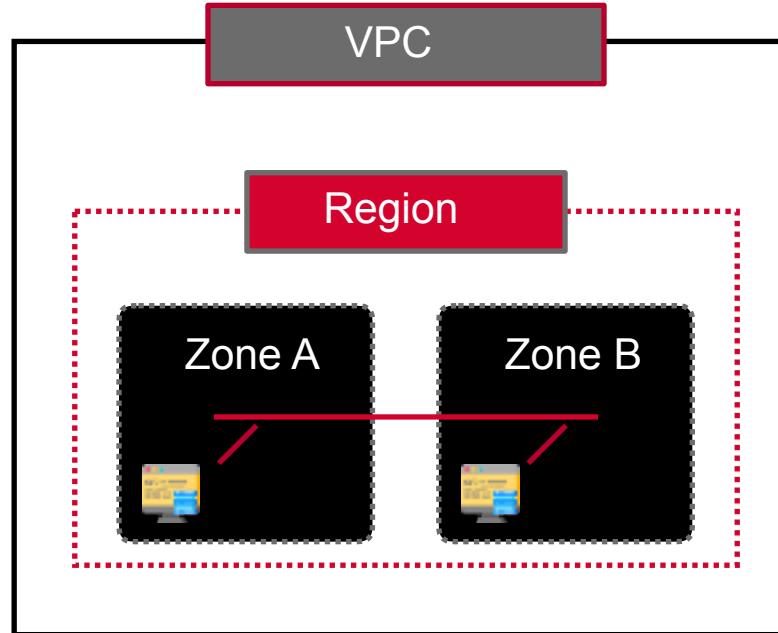
# Subnets

- **IP range partitions** within global VPCs
- VPCs have no IP ranges
- Subnets are regional - can span zones inside a region
- Network has to have at least one subnet before you can use it





# Subnets Span Zones





# Subnets and IP Ranges

- Each subnet must have primary address range
- Valid RFC 1918 CIDR block
- Subnet ranges in **same network cannot overlap**
- Subnet ranges in **different networks can overlap**





# Subnet IP Ranges

192.168.1.0/24



# Subnet IP Ranges

192.168.1.0/24



The suffix /24 determines how many bits of the IP address are reserved for the network portion versus the host portion



# Subnet IP Ranges

192.168.1.0/24



24 bits for the network  
8 bits for the hosts



# Valid IP Addresses in this Range

192.168.1.0/24



192.168.1.**1**

192.168.1.**67**

192.168.1.**128**

192.168.1.**255**



**In order to expand the IP addresses  
for a subnet assign a smaller portion  
for the base IP address of the network**



# Expand IP Addresses in Subnet

192.168.1.0/**20**



# AutoMode and CustomMode VPCs

## Auto Mode

Subnets automatically created  
in each region, default firewall  
rules

## Custom Mode

Manually create subnets in  
regions, no defaults  
preconfigured



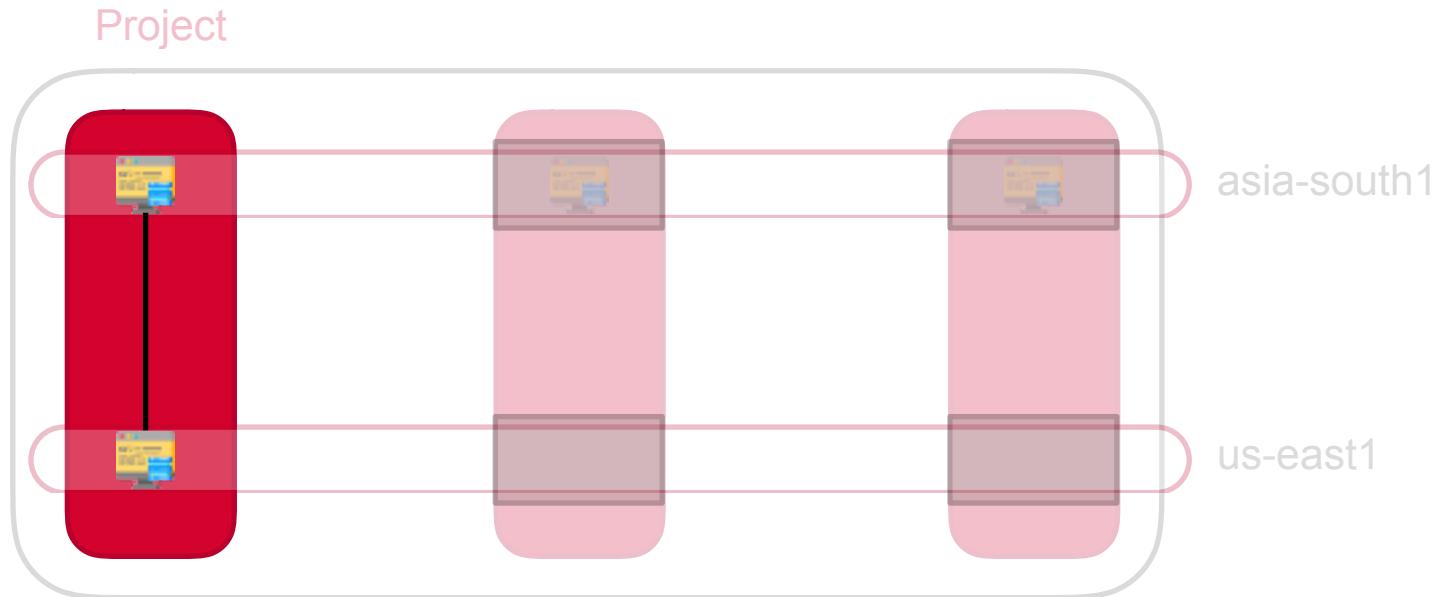
# AutoMode and CustomMode VPCs

- Auto Mode VPCs have pre-created subnets
  - One in each Google Cloud region
- Custom Mode VPCs start with no subnets
  - Full control over which regions have subnets
  - Can create multiple subnets in a region





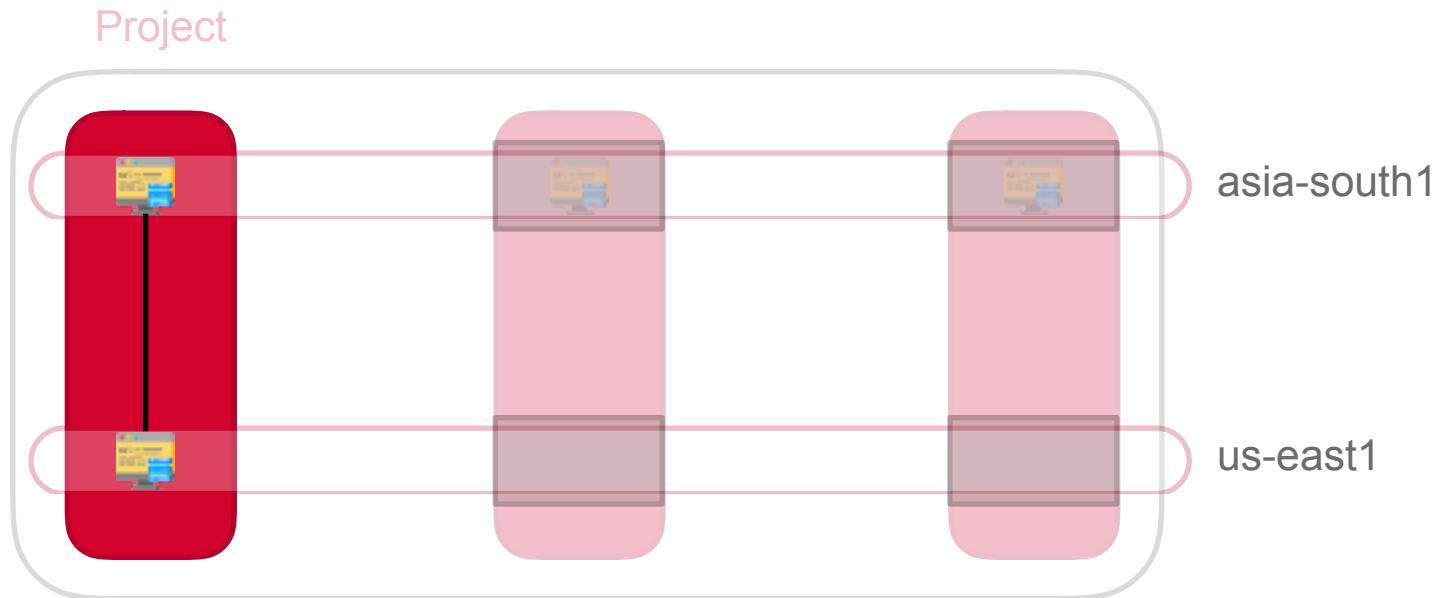
# Communication on VPCs



Resources within a VPC communicate using  
private IP addresses



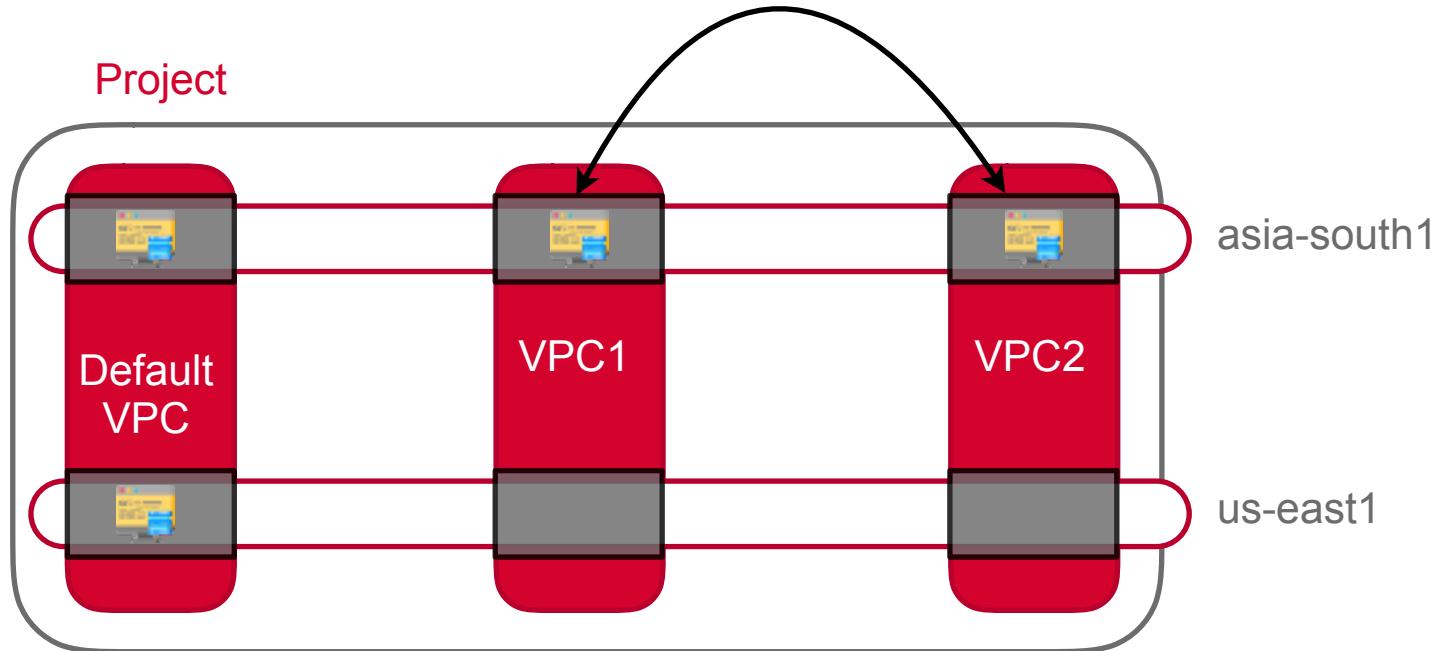
# Communication on VPCs



Wherever they are located in the world -  
irrespective of physical location



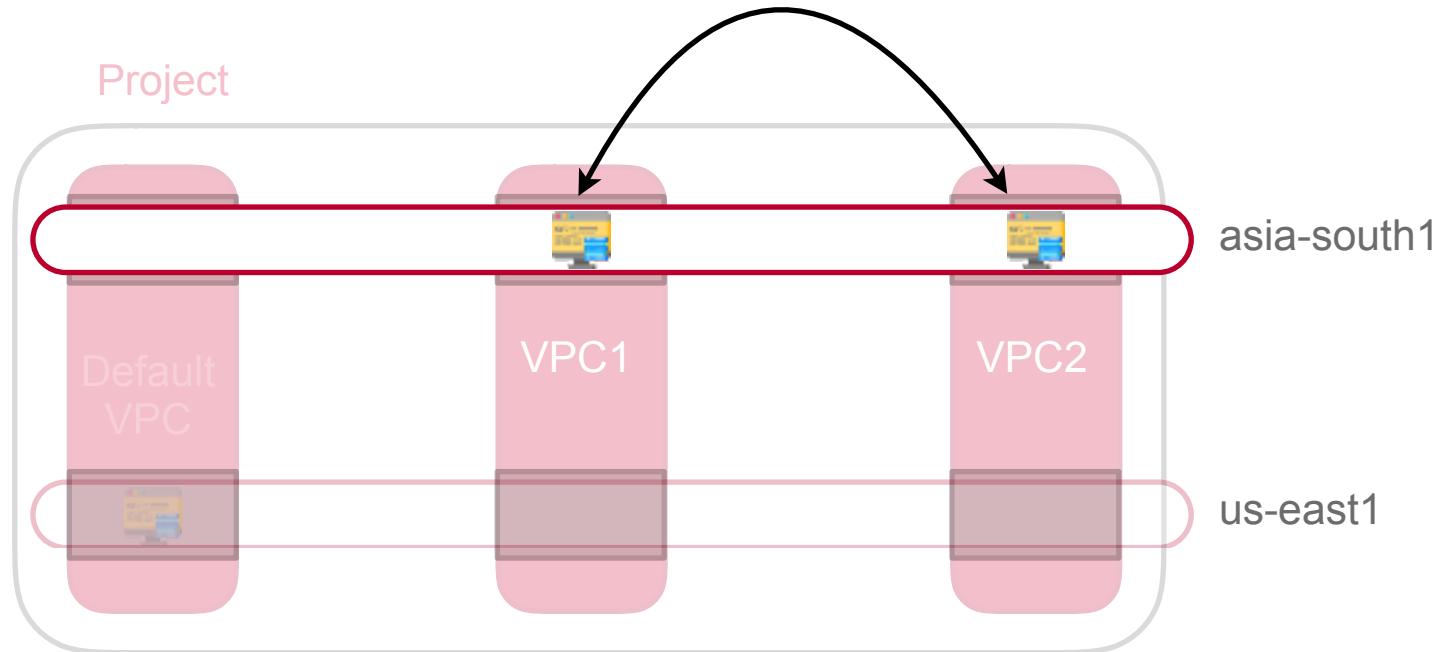
# Communication on VPCs



Resources on different VPCs communicate  
over the internet using external IPs



# Communication on VPCs



Even though they are in the same region - they may even be  
in the same zone on the same physical hardware



# Default VPC

- Pre-created on every project
- Includes subnet for each Google Cloud region
- New subnets added when new regions are created
- Resources created here by default





# Default VPC

- Includes routes for all resources
- All VMs on the default VPC can talk to each other
- Default gateway to internet
- Includes several firewall rules





# Firewall Rules

- Every VPC is a distributed firewall
- Firewall rules defined in VPC
- Are applied on per-instance basis
- Can also regulate internal traffic





# Firewall Rules

- Every VPC has two permanent rules
  - Implied **allow egress**
  - Implied **deny ingress**
- Can be overridden by more specific rules
- In addition, default VPC has several rules





# Additional Rules in Default VPC

- default-allow-internal
- default-allow-ssh
- default-allow-rdp
- Default-allow-icmp





# Ingress and Egress Firewall Rules

- Ingress firewall rules
  - Govern what can enter your network resources
- Egress firewall rules
  - Govern what can exit your network resources
- Ingress and egress are always from the point of view of the target



# Connecting Networks





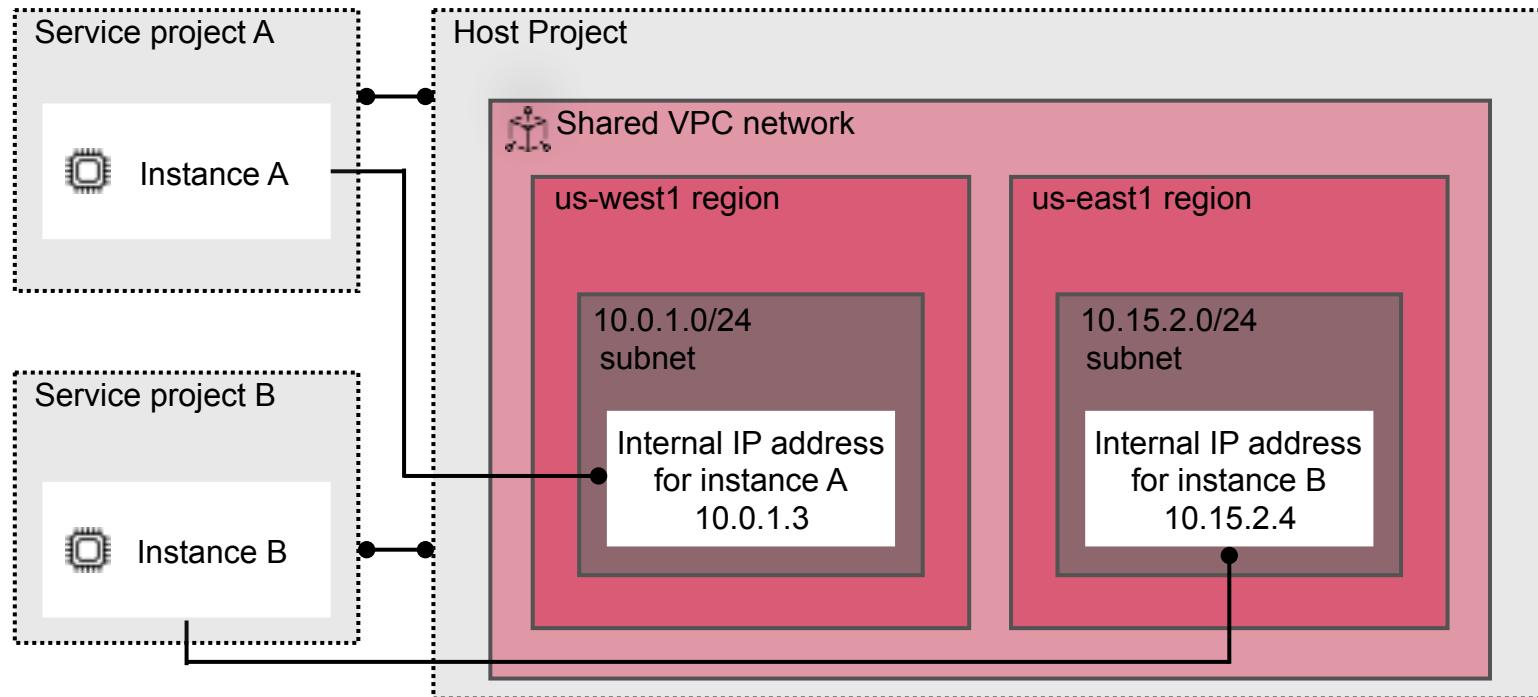
# Shared VPC

- Share VPC across projects on GCP
- One VPC shared across projects
- Projects must be in **the same organization**
- Host project, guest resources
- Shared VPC admin to administer the shared VPC





# Shared VPC





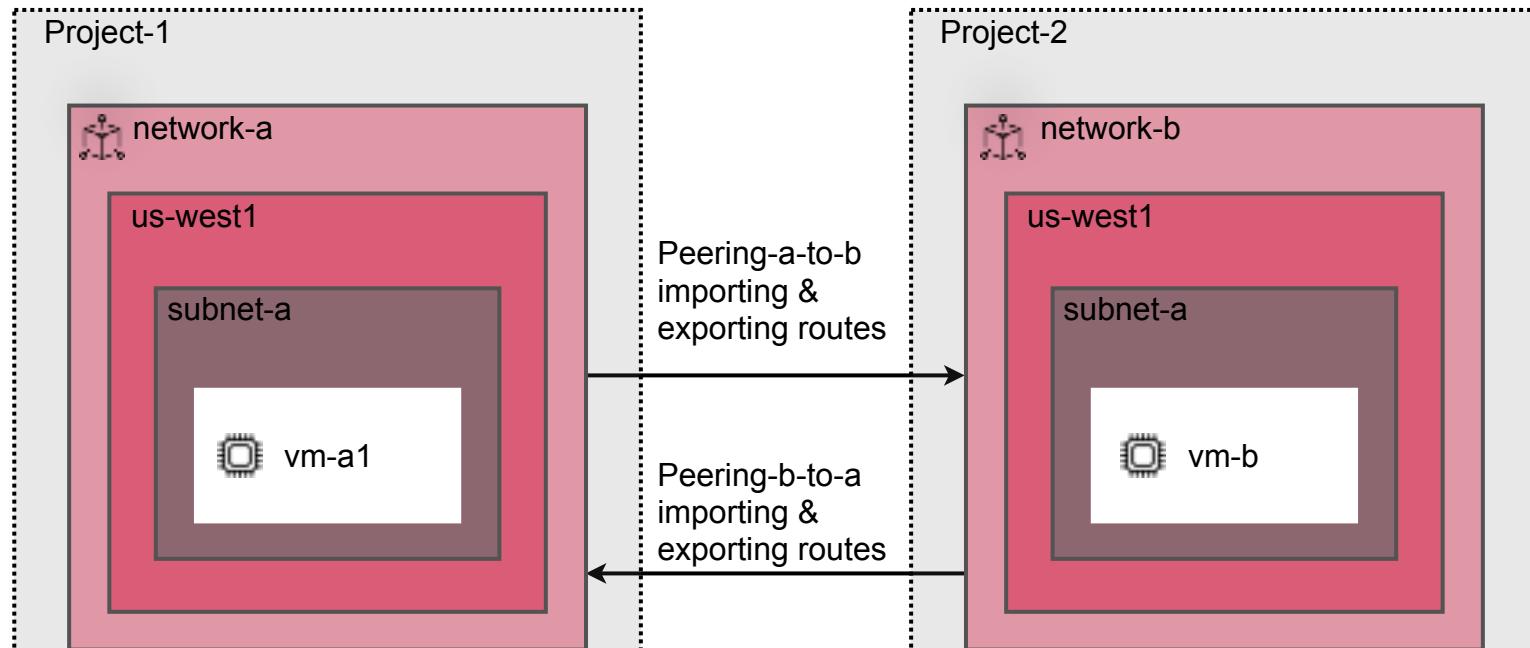
# VPC Peering

- Two or more VPCs shared across projects
- Projects need **not be in the same organization**
- Allows resources on different VPC networks to communicate using internal IP addresses
- Resources on the network use Google infrastructure to communicate
- Reduced latency, higher security and lower cost as compared with using external IPs





# VPC Peering





# Shared VPCs vs. Network Peering

## Shared VPCs

- Only within **same organization**
- One VPC used across projects
- Host and service projects not peers
- Single level of sharing possible

## Network Peering

- Across **organization boundaries**
- Multiple VPCs share resources
- Connected VPCs are peers
- Multiple levels of peering possible



# Interconnecting Networks

## GCP-to-GCP

VPC Network Peering

## Enterprise connectivity

Peering and interconnect options



# Interconnecting Networks

## GCP-to-GCP

VPC Network Peering

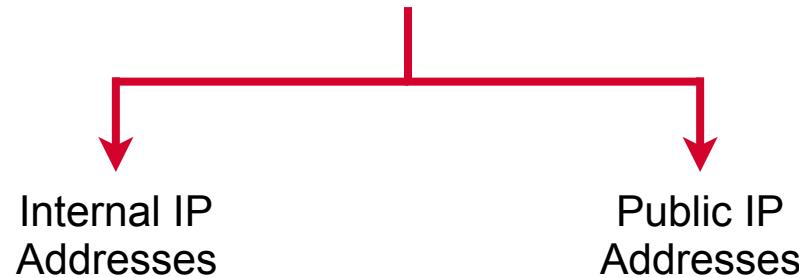
## Enterprise connectivity

Peering and interconnect options

Connect a cloud network with an on-premise network using private or public IP addresses

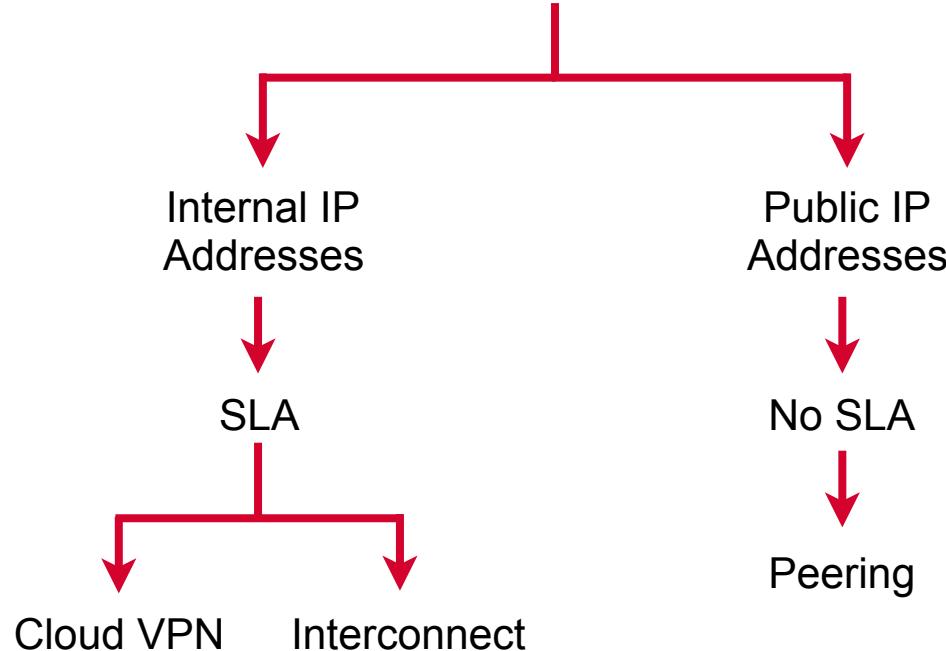


# Enterprise Connectivity



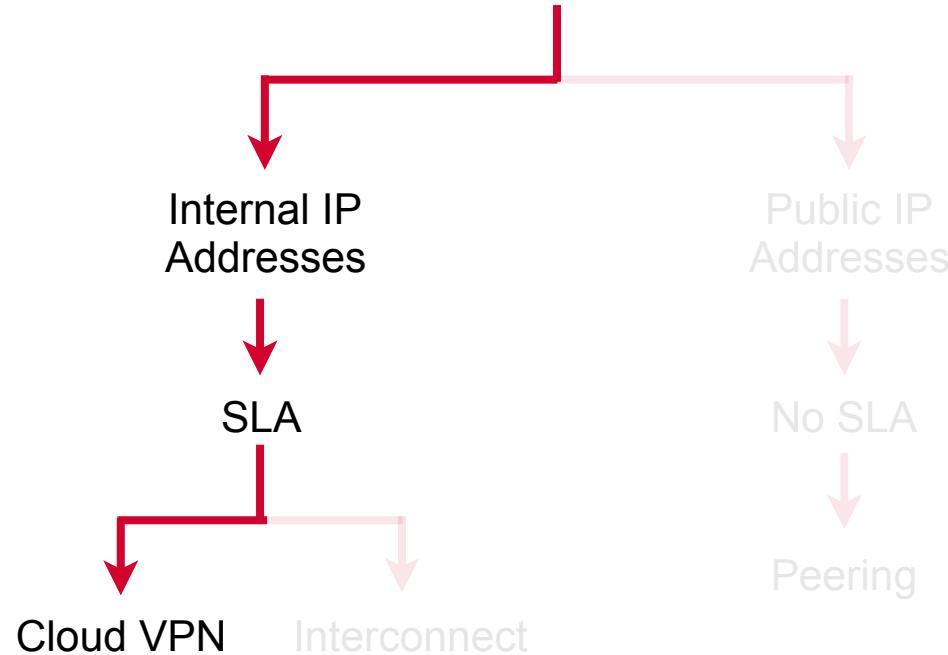


# Enterprise Connectivity





# Enterprise Connectivity





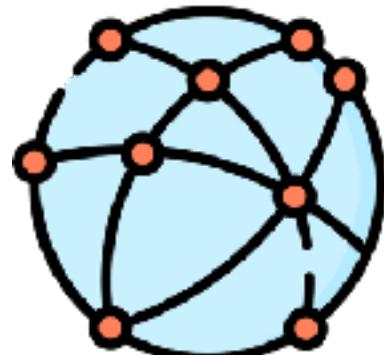
# Cloud VPN

Configuration Property	Choice
Connection	<b>Encrypted tunnel to VPC networks through the public internet</b>
Access Type	<b>Internal IP addresses in RFC 1918 address space</b>
Capacity	<b>1.5-3 Gbps for each tunnel</b>
Other Considerations	<b>Requires a VPN device on your on-premises network</b>

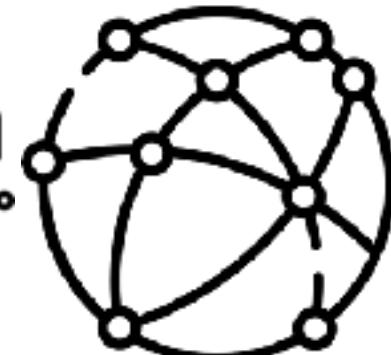


# Cloud VPN

Cloud Network



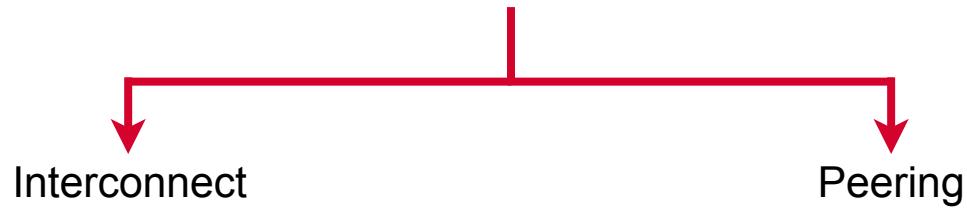
On-prem Network



- Two VPN gateways
- One for cloud network, another for on-prem network
- Traffic encrypted at one gateway
- Decrypted at other gateway
- Keys need to be exchanged

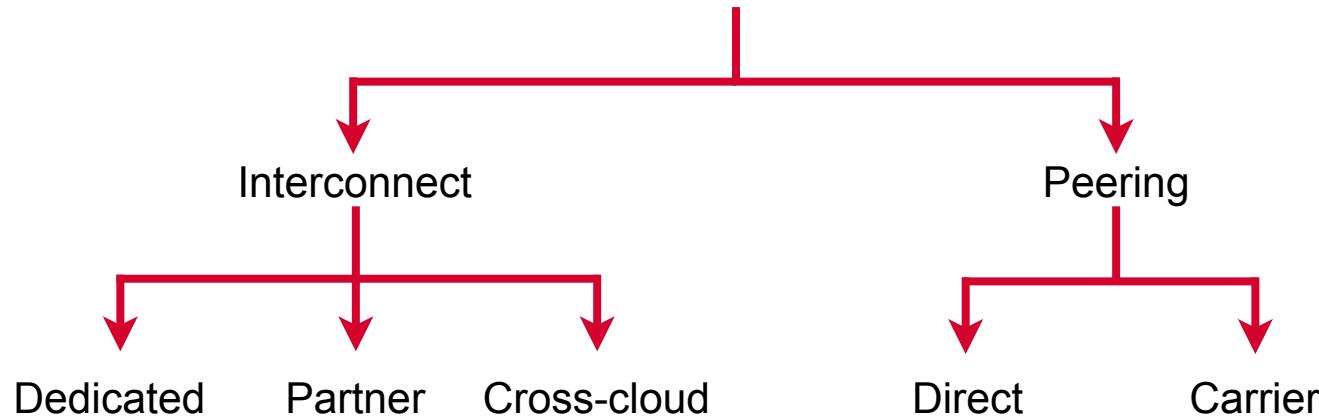


# Enterprise Connectivity



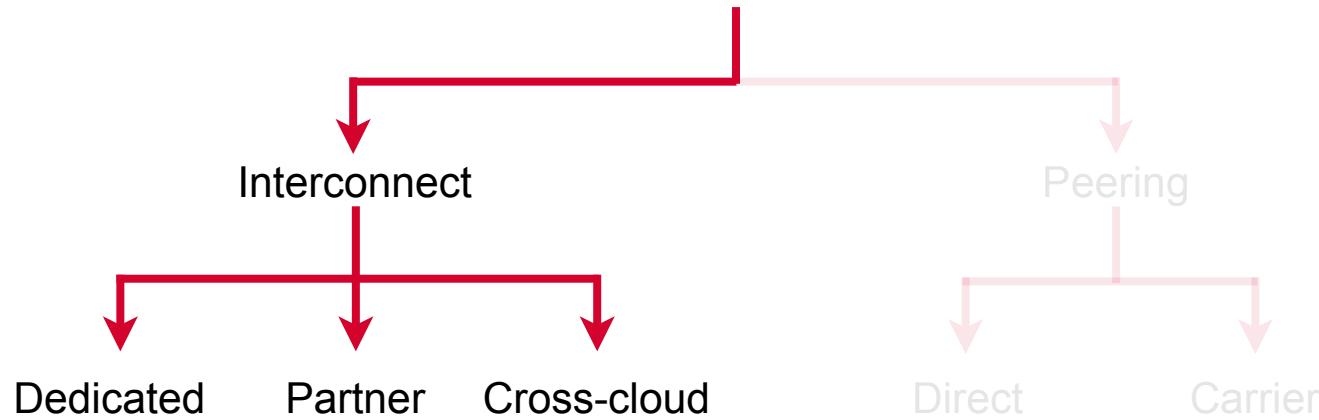


# Enterprise Connectivity



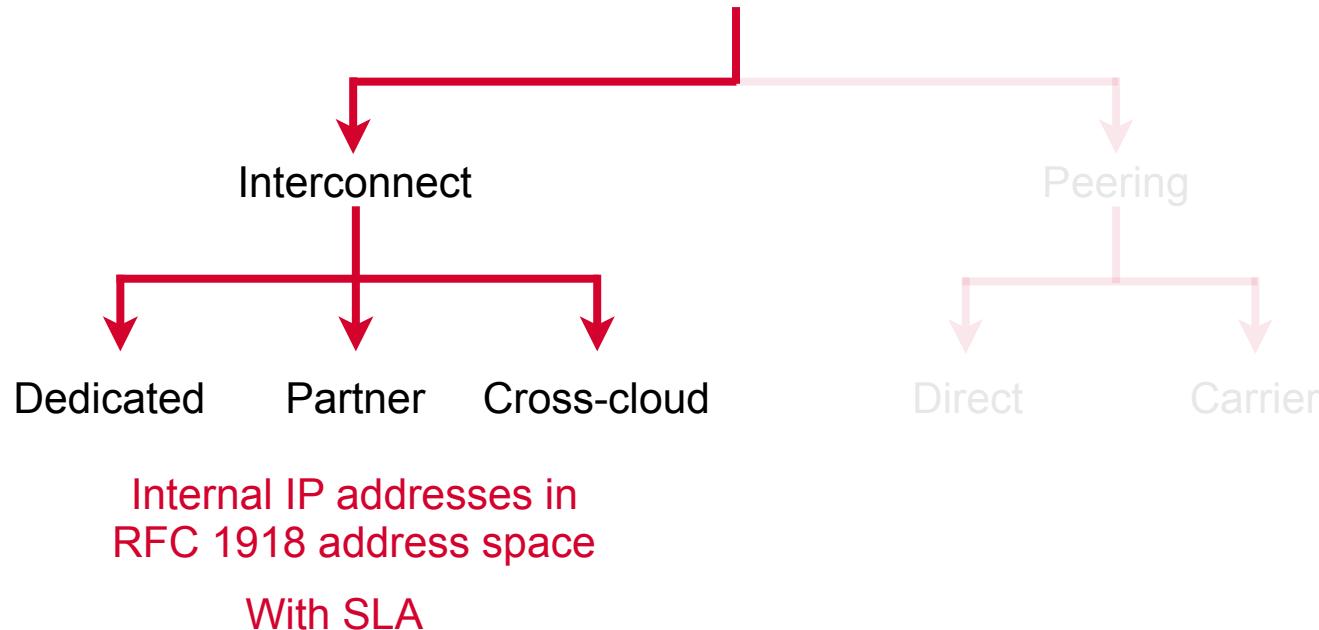


# Enterprise Connectivity





# Enterprise Connectivity





**Traffic between your external  
network and Google network DOES  
NOT traverse the public internet**



# Dedicated Interconnect

Configuration Property	Choice
Connection	Dedicated, direct connection to VPC networks
Access Type	Internal IP addresses in RFC 1918 address space
Capacity	10 Gbps or 100 Gbps connections
Other Considerations	<b>Must have connection in a Google supported colocation facility that supports the regions you want to connect to</b>



# Partner Interconnect

Configuration Property	Choice
Connection	Dedicated Bandwidth, connection to VPC network through a service provider
Access Type	Internal IP addresses in RFC 1918 address space
Capacity	50Mbps - 50Gbps per connection
Other Considerations	<b>Service providers might have specific restrictions or requirements</b>



# Cross-cloud Interconnect

- High-bandwidth dedicated connectivity between Google Cloud and another service provider
- Google will provision a dedicated physical connection
- Useful for:
  - Site-to-site data transfer
  - Multi-cloud strategy





# Cross-cloud Interconnect

Configuration Property	Choice
Connection	Dedicated physical connection between Google Cloud and other cloud platform
Access Type	Internal IP addresses in RFC 1918 address space
Capacity	10 Gbps or 100Gbps
Other Considerations	<b>Supported cloud providers</b> <b>AWS, Azure, Oracle, Alibaba</b>



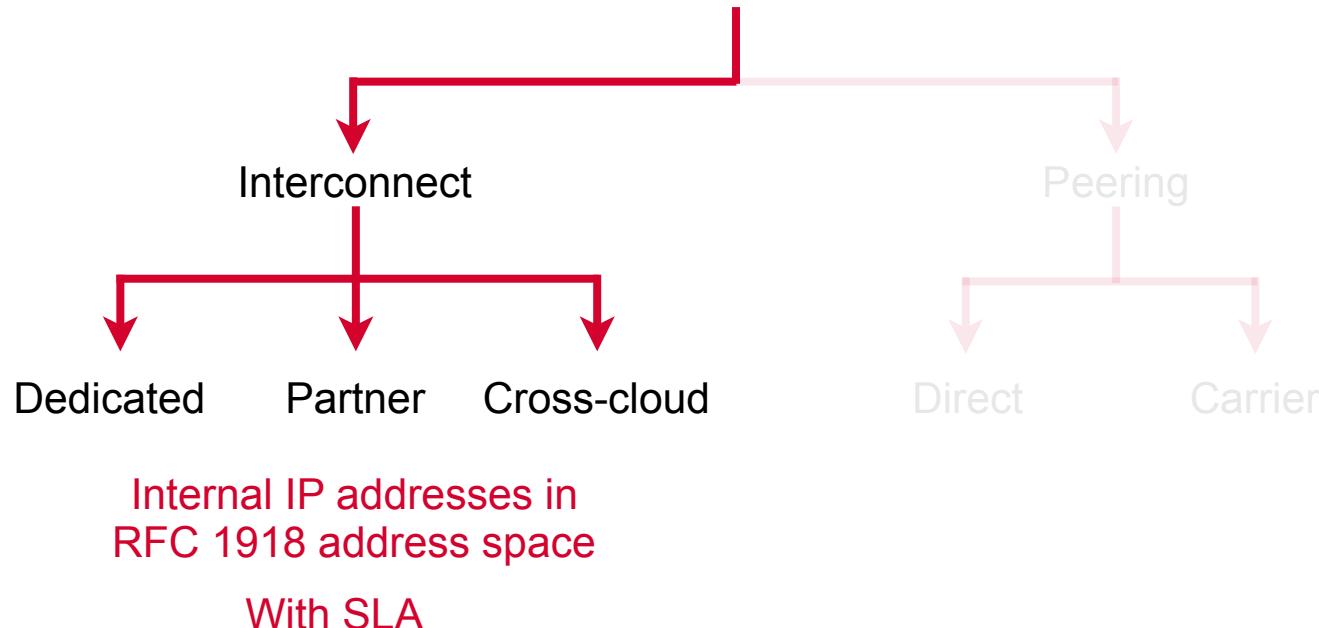
# Cloud Router

- Cloud Router is a fully distributed and managed Google Cloud service that **dynamically manages routing tables**
- Uses the Border **Gateway Protocol (BGP)** to exchange routes between Google Cloud and on-premise networks
- Allows for **automatic updation** when network changes occur
- Used with Cloud Interconnect and Cloud VPN



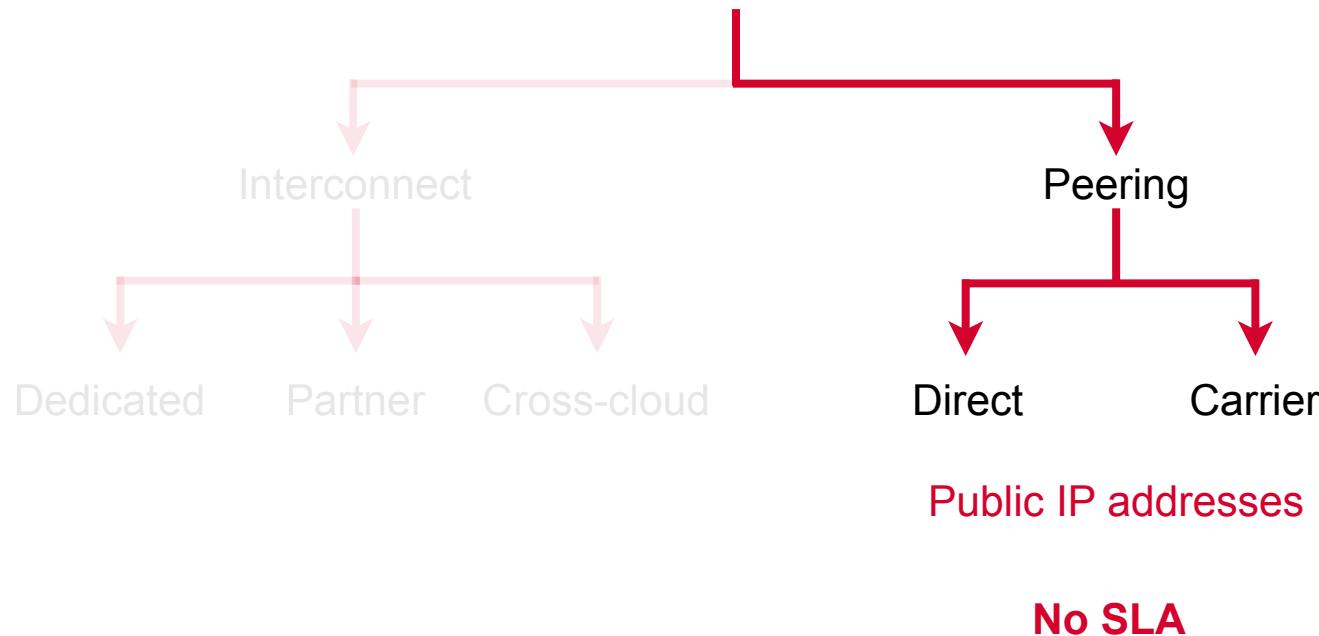


# Enterprise Connectivity





# Enterprise Connectivity





# Direct Peering

Configuration Property	Choice
Connection	Provides direct access from your on-premises network to Google Workspace and Google APIs for the full suite of Google Cloud products.
Access Type	Public IP addresses
Other Considerations	<b>Connects to Google's edge network</b>



# Carrier Peering

Configuration Property	Choice
Connection	Peering through service provider to access Google applications such as Google Workspace and to Google Cloud products that can be exposed through one or more public IP addresses.
Access Type	Public IP addresses
Other Considerations	<b>Connects to Google's edge network through a service provider. Requirements vary by partner</b>

# VPCs

Your company has an application server running on one subnet with the network tag "app-server" and a database server running on another subnet with the network tag "db-server" within the same Google Cloud VPC. You want to configure a firewall rule to allow traffic only from the application server to the database server, using network tags, and prevent access from any other sources. What should you do?

- A. Configure an ingress firewall rule that allows traffic from the "app-server" tag to the "db-server" tag.
- B. Create an egress firewall rule that allows traffic from the "app-server" subnet to the "db-server" subnet.
- C. Set up an ingress firewall rule that allows traffic from all sources to the "db-server" tag, restricted by port.
- D. Implement an egress firewall rule that allows traffic from the "db-server" tag to the "app-server" tag, restricted by IP address.



# VPCs

Your company has an application server running on one subnet with the network tag "app-server" and a database server running on another subnet with the network tag "db-server" within the same Google Cloud VPC. You want to configure a firewall rule to allow traffic only from the application server to the database server, using network tags, and prevent access from any other sources. What should you do?

- A. **Configure an ingress firewall rule that allows traffic from the "app-server" tag to the "db-server" tag.**
- B. Create an egress firewall rule that allows traffic from the "app-server" subnet to the "db-server" subnet.
- C. Set up an ingress firewall rule that allows traffic from all sources to the "db-server" tag, restricted by port.
- D. Implement an egress firewall rule that allows traffic from the "db-server" tag to the "app-server" tag, restricted by IP address.



# VPCs

Your team is managing a Google Cloud project with a VPC in custom mode. All the available IP address ranges for the existing subnets have been used up, and you need to allocate more IP addresses for future resources in the project. What should you do to allow for more IP addresses?

- A. Create a new VPC in auto mode to automatically allocate new subnets with additional IP ranges.
- B. Resize the existing subnets to increase the available IP address range within the same VPC.
- C. Add a new subnet with a non-overlapping IP range within the same VPC to expand the available IP addresses.
- D. Switch the VPC to auto mode to automatically adjust and assign additional IP ranges to the existing subnets.



# VPCs

Your team is managing a Google Cloud project with a VPC in custom mode. All the available IP address ranges for the existing subnets have been used up, and you need to allocate more IP addresses for future resources in the project. What should you do to allow for more IP addresses?

- A. Create a new VPC in auto mode to automatically allocate new subnets with additional IP ranges.
- B. Resize the existing subnets to increase the available IP address range within the same VPC.**
- C. Add a new subnet with a non-overlapping IP range within the same VPC to expand the available IP addresses.
- D. Switch the VPC to auto mode to automatically adjust and assign additional IP ranges to the existing subnets.



# VPCs

Your team is managing a Google Cloud project with a VPC in custom mode. All the available IP address ranges for the existing subnets have been used up, and you need to allocate more IP addresses for future resources in the project. What should you do to allow for more IP addresses?

- A. Create a new VPC in auto mode to automatically allocate new subnets with additional IP ranges.
- B. Resize the existing subnets to increase the available IP address range within the same VPC.
- C. Add a new subnet with a non-overlapping IP range within the same VPC to expand the available IP addresses.
- D. Switch the VPC to auto mode to automatically adjust and assign additional IP ranges to the existing subnets.



# VPCs

Your team is managing a Google Cloud project with a VPC in custom mode. All the available IP address ranges for the existing subnets have been used up, and you need to allocate more IP addresses for future resources in the project. What should you do to allow for more IP addresses?

- A. Create a new VPC in auto mode to automatically allocate new subnets with additional IP ranges.
- B. Resize the existing subnets to increase the available IP address range within the same VPC.**
- C. Add a new subnet with a non-overlapping IP range within the same VPC to expand the available IP addresses.
- D. Switch the VPC to auto mode to automatically adjust and assign additional IP ranges to the existing subnets.



# VPCs

Your company has two separate teams, each managing its own Google Cloud project. Team A operates in **project-alpha** with VPC **vpc-alpha** (192.168.0.0/16), and Team B works in **project-beta** with VPC **vpc-beta** (172.16.0.0/16). Team A's application needs to communicate with services managed by Team B **using internal IP addresses**. You want to enable secure communication between the two VPCs in a cost-effective manner while adhering to Google's best practices.

What should you do?

- A. Configure Cloud NAT to enable communication between the VPCs.
- B. Set up VPN network peering between the two VPCs to establish direct communication using internal IPs.
- C. Deploy a bastion host in one of the VPCs to serve as a bridge for network traffic.
- D. Use Public IP addresses to allow the services to communicate over the internet.



# VPCs

Your company has two separate teams, each managing its own Google Cloud project. Team A operates in **project-alpha** with VPC **vpc-alpha** (192.168.0.0/16), and Team B works in **project-beta** with VPC **vpc-beta** (172.16.0.0/16). Team A's application needs to communicate with services managed by Team B **using internal IP addresses**. You want to enable secure communication between the two VPCs in a cost-effective manner while adhering to Google's best practices.

What should you do?

- A. Configure Cloud NAT to enable communication between the VPCs.
- B. Set up VPN network peering between the two VPCs to establish direct communication using internal IPs.**
- C. Deploy a bastion host in one of the VPCs to serve as a bridge for network traffic.
- D. Use Public IP addresses to allow the services to communicate over the internet.



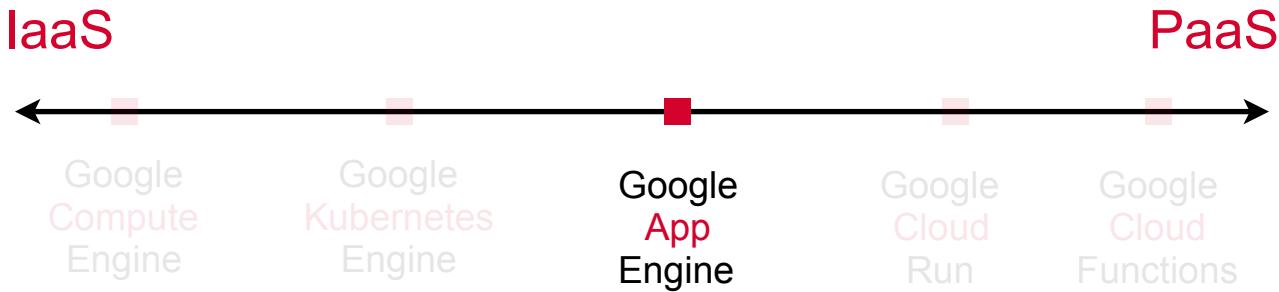
O'REILLY®

# Google App Engine





# Google Cloud Compute Choices





# Google App Engine

Web framework and platform for hosting web applications on the Google Cloud

Support for Go, PHP, Java, Python, Node.js, .NET, Ruby and other languages



# Google App Engine

Web framework and platform for hosting web applications on the Google Cloud

Support for Go, PHP, Java, Python, Node.js, .NET, Ruby and other languages

Focus on development and code

Infrastructure and scaling taken care of by the platform



# App Engine Environments

**Standard Environment**

**Flexible Environment**



# App Engine Environments

## Standard

- App runs in a **proprietary sandbox**
- Instances start up in seconds
- Code in few languages/versions only
- No other runtimes possible
- Apps cannot access Compute Engine resources
- **Can install 3rd party binaries only for selected runtimes**



# App Engine Environments

## Standard

- App runs in a **proprietary sandbox**
- Instances start up in seconds
- Code in few languages/versions only
- No other runtimes possible
- Apps cannot access Compute Engine resources
- **Can install 3rd party binaries only for selected runtimes**

## Flexible

- Runs in **Docker container** on GCE VM
- Instance start up in minutes
- Code in far more languages/versions
- **Custom runtimes possible**
- Apps can access Compute Engine resources, some OS packages
- Can install and access third-party binaries



# App Engine Environments

## Standard

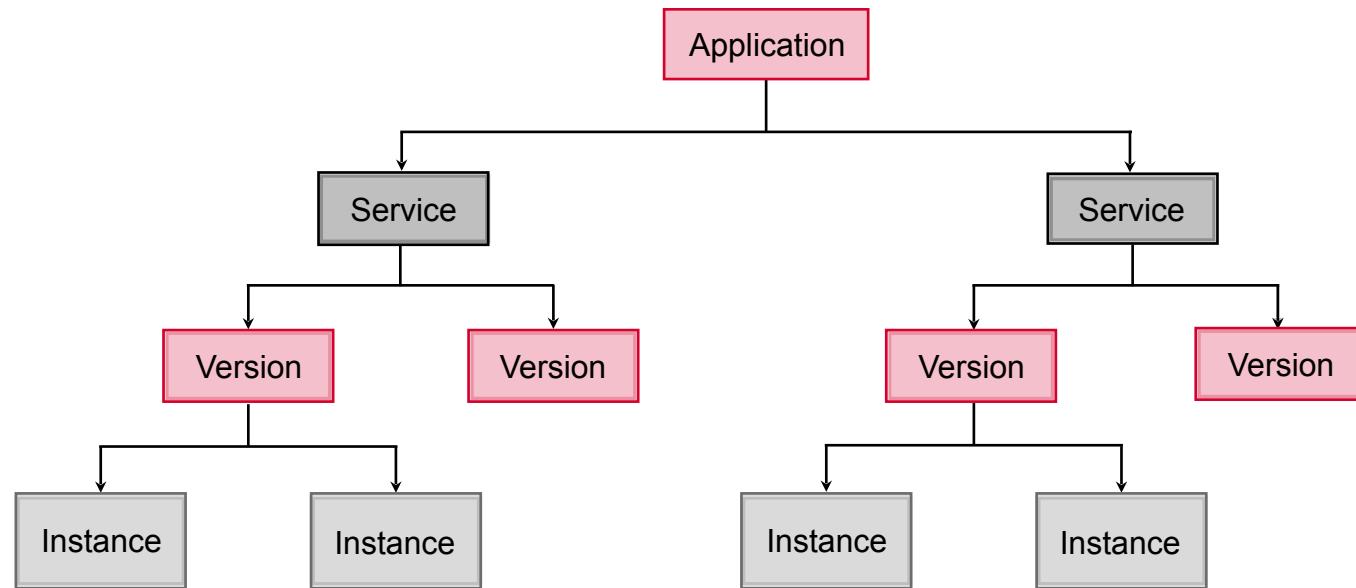
- Apps that experience **traffic spikes**
- Usually **stateless** HTTP web apps

## Flexible

- Apps that experience **consistent traffic**
- General purpose apps

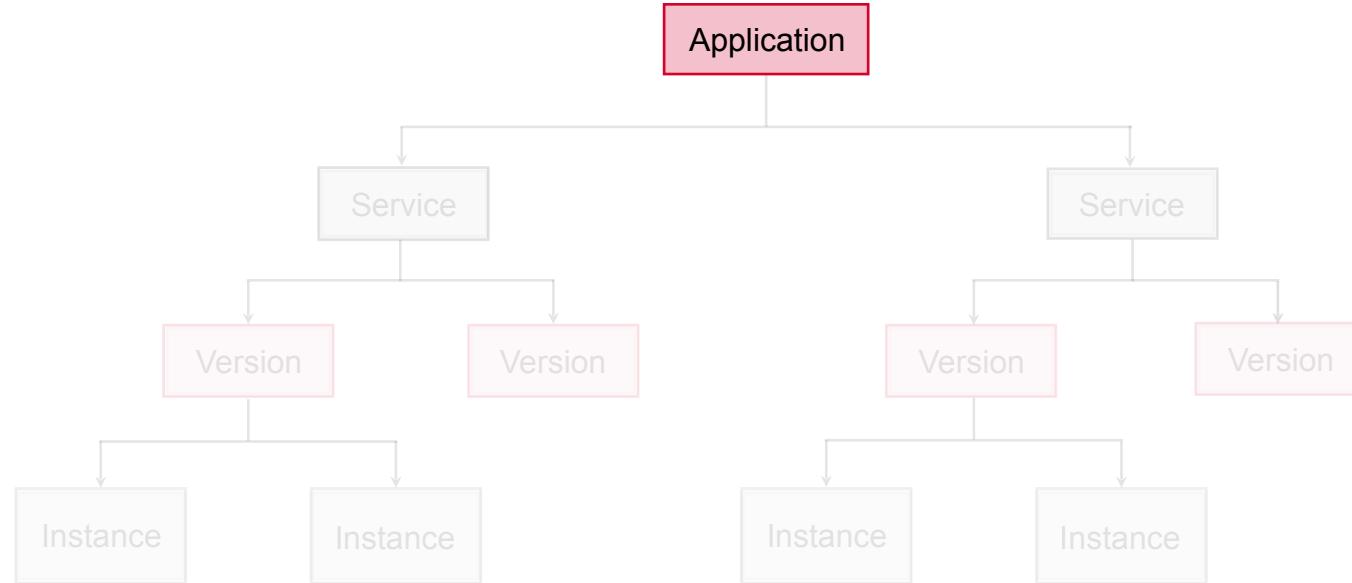


# App Engine App





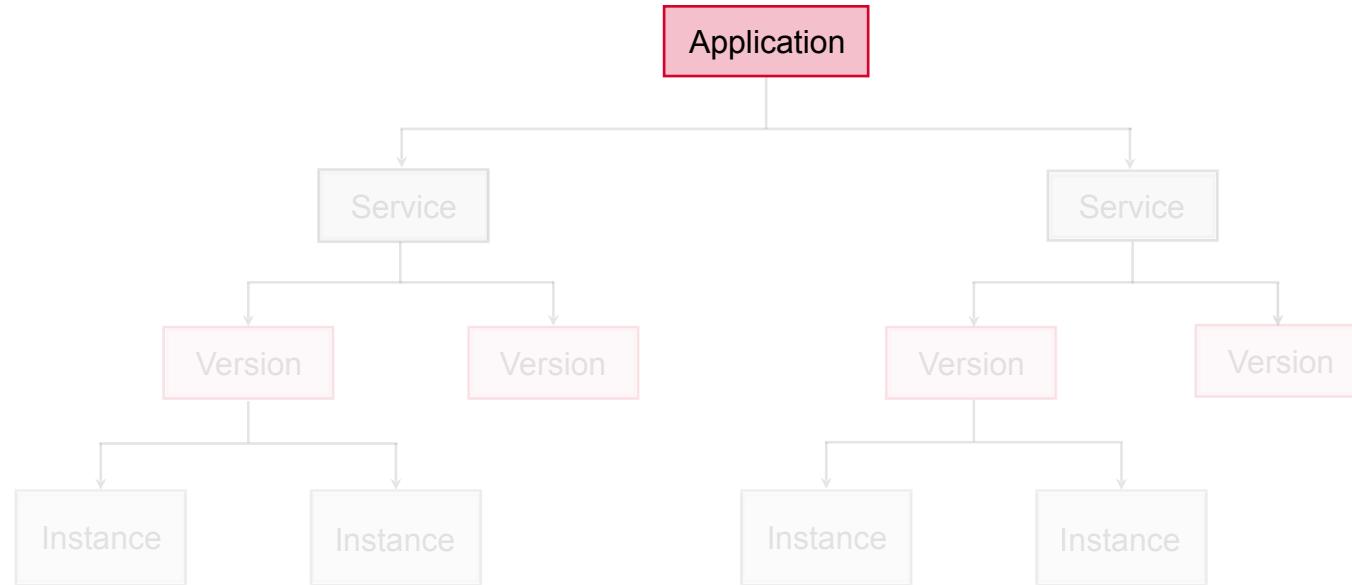
# Application



Top-level container for multiple services, their versions, and instances



# Application



Only one App Engine app can be provisioned in each project

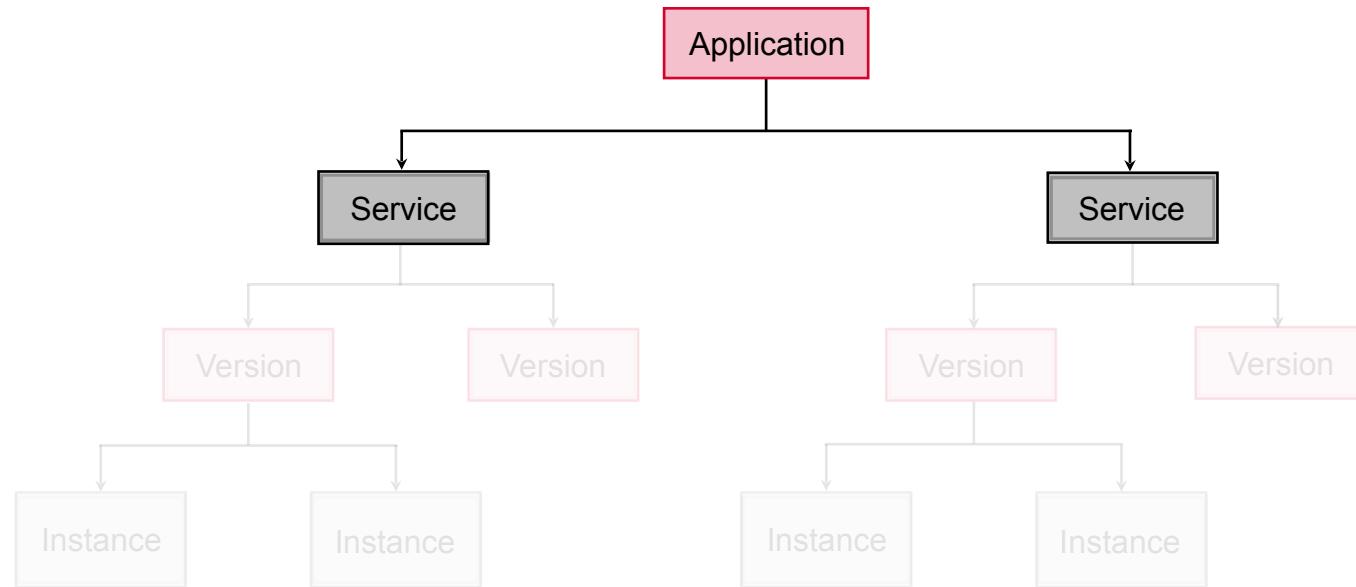


# Service

Each service in an App Engine app behaves like a microservice and scales independently



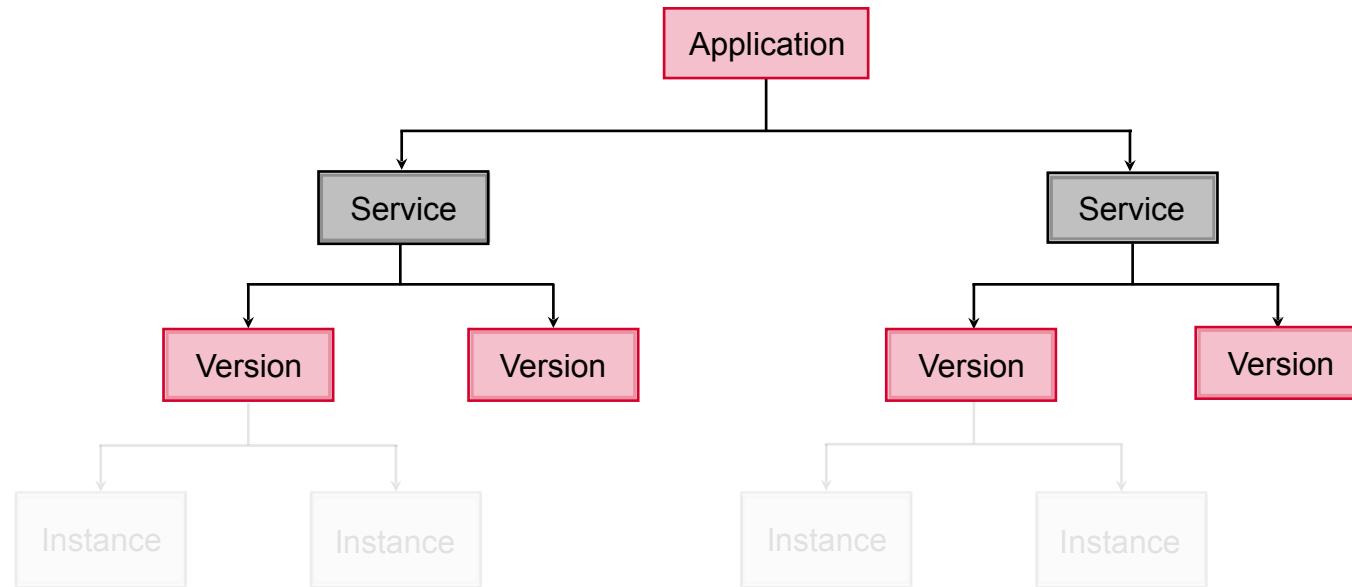
# Service



Every app includes at least one service - the **default** service



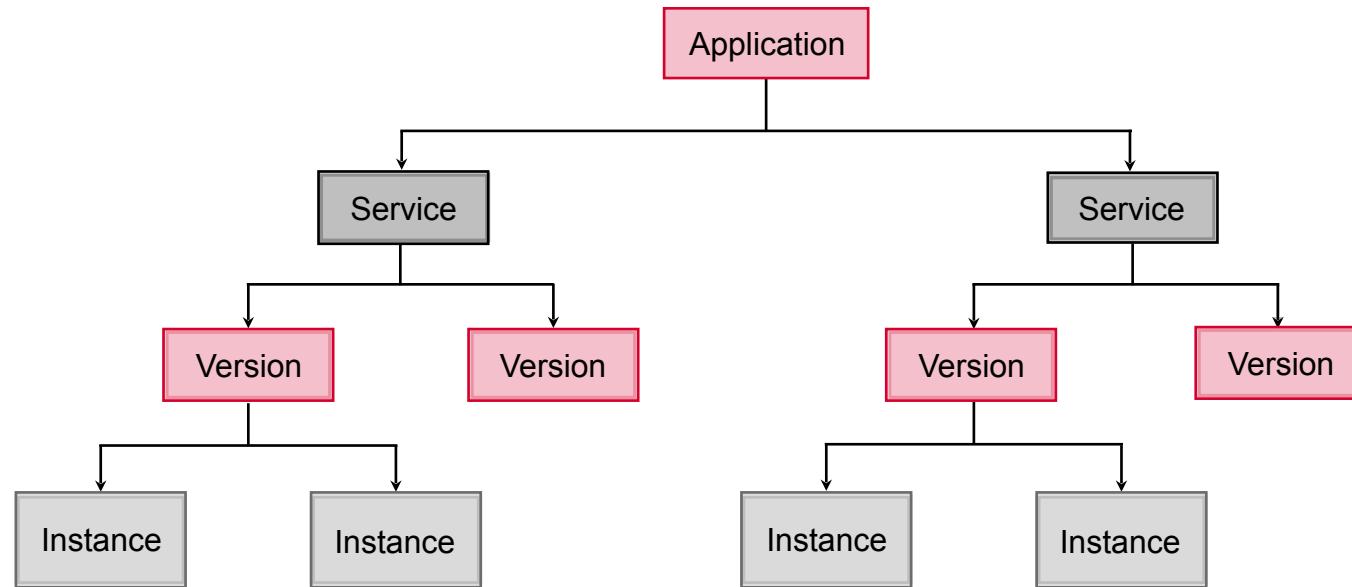
# Service Versions



Every new deployment to a particular service creates a new version of the service



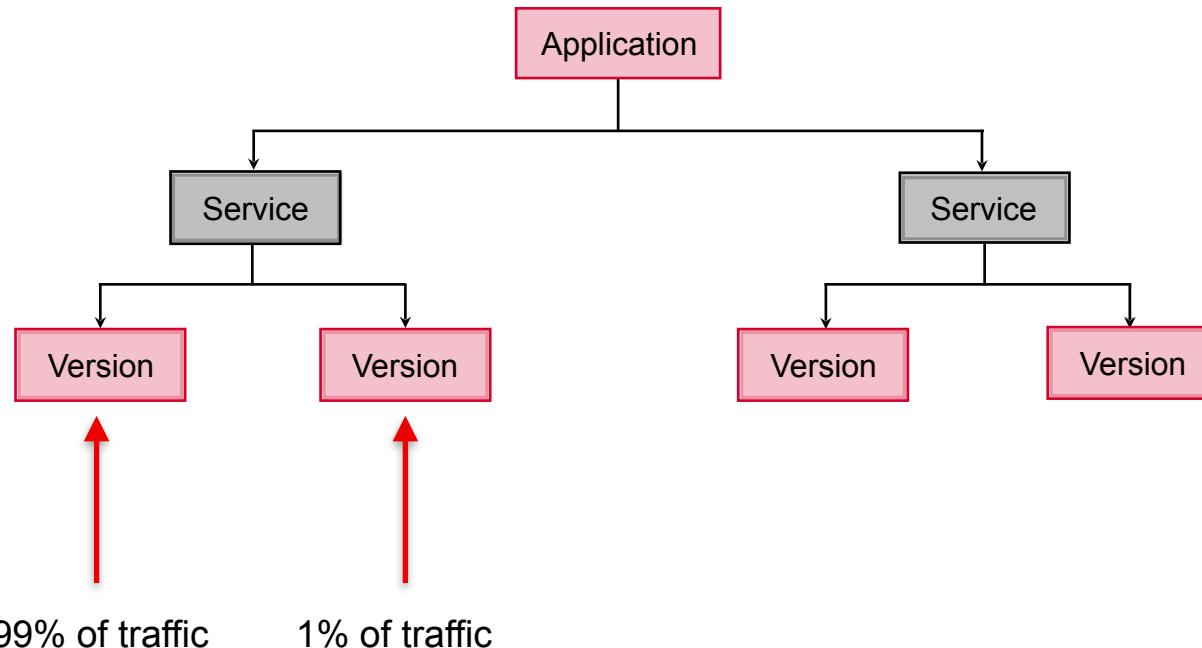
# Instances



Versions run on one or more instances - can configure App Engine to automatically scale instance based on load



# Traffic Splitting



Can split traffic between versions to gradually ramp up traffic to the new version - allows you to conduct A/B testing between versions



# Traffic Splitting

- **Weighted percentages splitting**
  - 70% to version 1, 30% to version 2
- **Cookie-based splitting**
  - Splitting based on session cookie
  - Once user is assigned a new version they are consistently routed to the new version
- **IP address splitting**
  - Users from the same IP range will be directed to the same version



# App Engine

Your team is planning to release a new feature for an existing App Engine application and wants to perform A/B testing to compare the performance of the new feature against the current version. The goal is to send a portion of your users to the new version and keep the rest on the stable version, allowing you to measure the impact and gather data before fully rolling out the feature. You also want to ensure a seamless rollback if issues arise, without adding development or operational complexity for your users. What should you do?

- A. Manually ask users to switch between different versions of the application to gather feedback.
- B. Use App Engine traffic splitting to send a portion of the traffic to the new version while keeping the rest on the stable version for A/B testing.
- C. Release the new version to all users and revert to the old version if feedback is negative.
- D. Set up two separate App Engine services and instruct users to alternate between them for A/B testing.



# App Engine

Your team is planning to release a new feature for an existing App Engine application and wants to perform A/B testing to compare the performance of the new feature against the current version. The goal is to send a portion of your users to the new version and keep the rest on the stable version, allowing you to measure the impact and gather data before fully rolling out the feature. You also want to ensure a seamless rollback if issues arise, without adding development or operational complexity for your users. What should you do?

- A. Manually ask users to switch between different versions of the application to gather feedback.
- B. Use App Engine traffic splitting to send a portion of the traffic to the new version while keeping the rest on the stable version for A/B testing.**
- C. Release the new version to all users and revert to the old version if feedback is negative.
- D. Set up two separate App Engine services and instruct users to alternate between them for A/B testing.

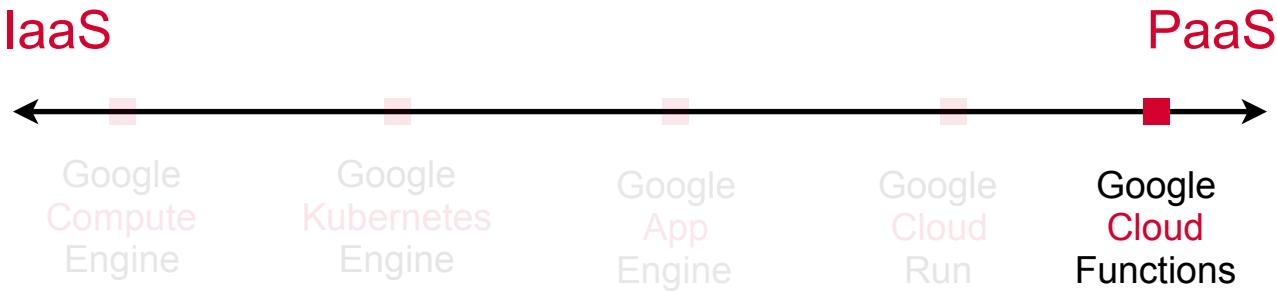


# Google Cloud Run Functions





# Google Cloud Compute Choices



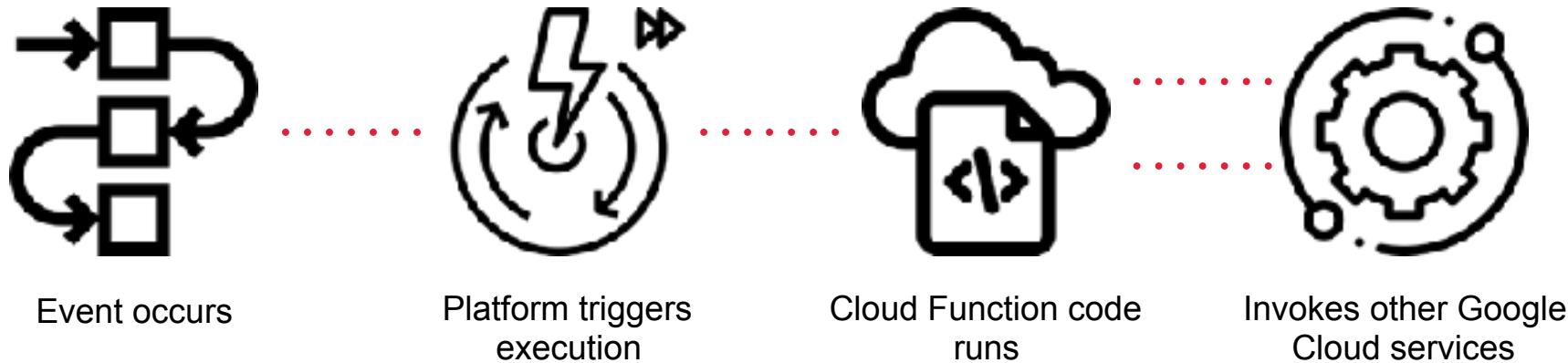


# Cloud Functions

Event-driven serverless compute platform

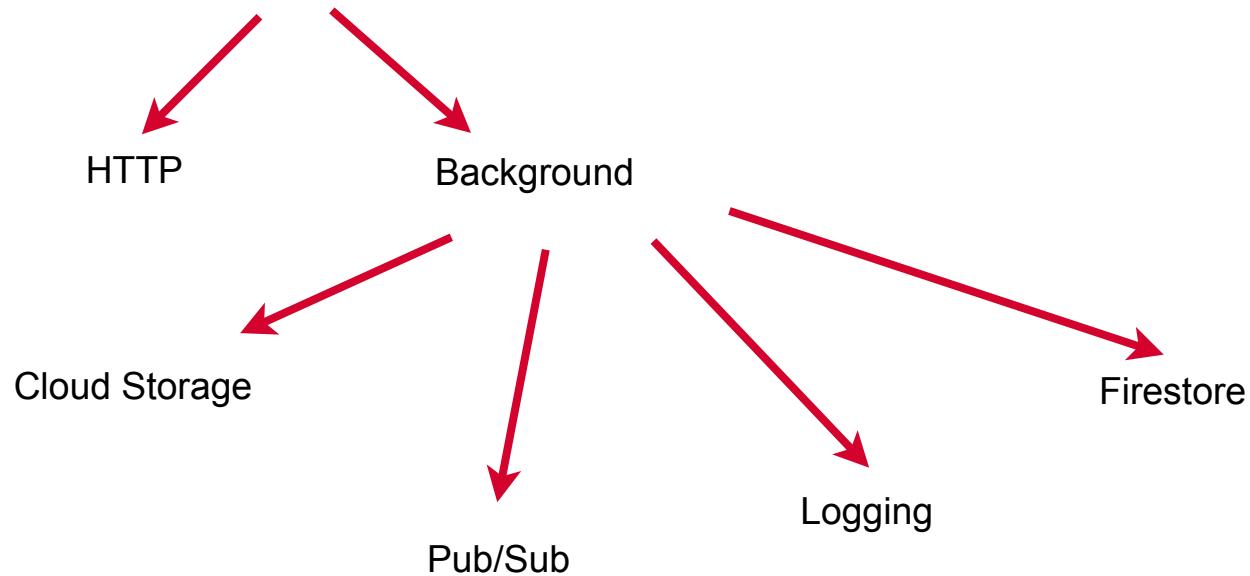


# Event-driven Serverless Compute





# Types of Events





# Concurrency and Scale

- Spin up function instances based on current load
- Functions receive event parameters from platform
- Functions do not share memory or variables
- An instance processes a single request (generation 1)
- Function concurrency supported (generation 2)
- Functions should be **stateless**

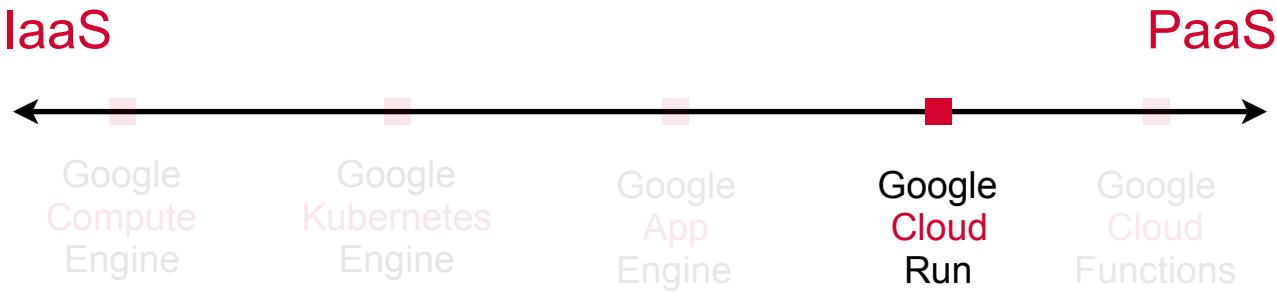


# Google Cloud Run





# Google Cloud Compute Choices





# Container

A container image is a lightweight, stand-alone, executable package of a piece of **software that includes everything needed to run it**; code, runtime, system tools, system libraries, settings



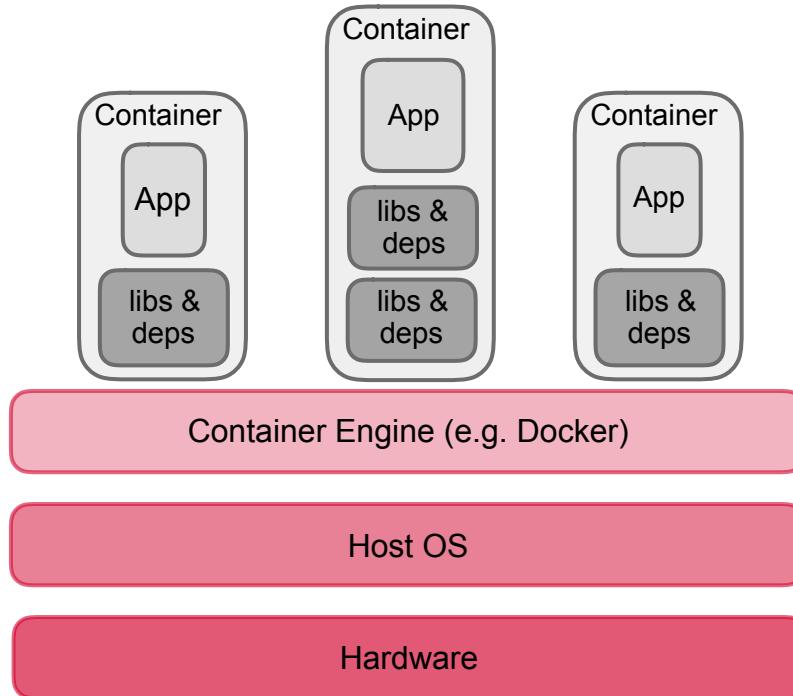
# Container

- Contains applications
- And all of the application's dependencies
- Platform independent
- Runs on layer of abstraction
- Docker Runtime (for Docker containers)





# Modern Workloads on Containers





# Cloud Run

Serverless, managed platform that lets you run containers directly on top of Google's scalable architecture



# Cloud Run

- Write your code in any programming language
- Create a container image (or use source-based deployment option - Google Cloud will build container image for you)
- Register the container with the Artifact Registry
- Deploy your container directly using Cloud Run
- **No cluster creation no infrastructure management**
- Request-based pricing and instance-based pricing





# Running Code Using Cloud Run

Cloud Run Services

Cloud Run Jobs

Both use the same environment and have the same integrations with other Google Cloud services



# Cloud Run Services

- Used to run code that responds to web requests or events
- Each service located in a Google Cloud region
- Replicated across zones in the region
- Exposes an endpoint
- **Automatically scales underlying infrastructure to handle incoming requests**
- **Version management, rollbacks, traffic management - all handled by the platform**





# Cloud Run Jobs

- Used to run code that performs work (a job) and quits when the job is done
- Each service located in a Google Cloud region and executes one or more containers to completion
- A job comprises of many tasks executing in parallel - each container runs one task





# Cloud Run Revisions

Represents different versions of your deployed application. Each new version of your application is deployed as a new **revision**.



# Cloud Run Revisions

Website-v4

Website-v3

Website-v2

Website-v1



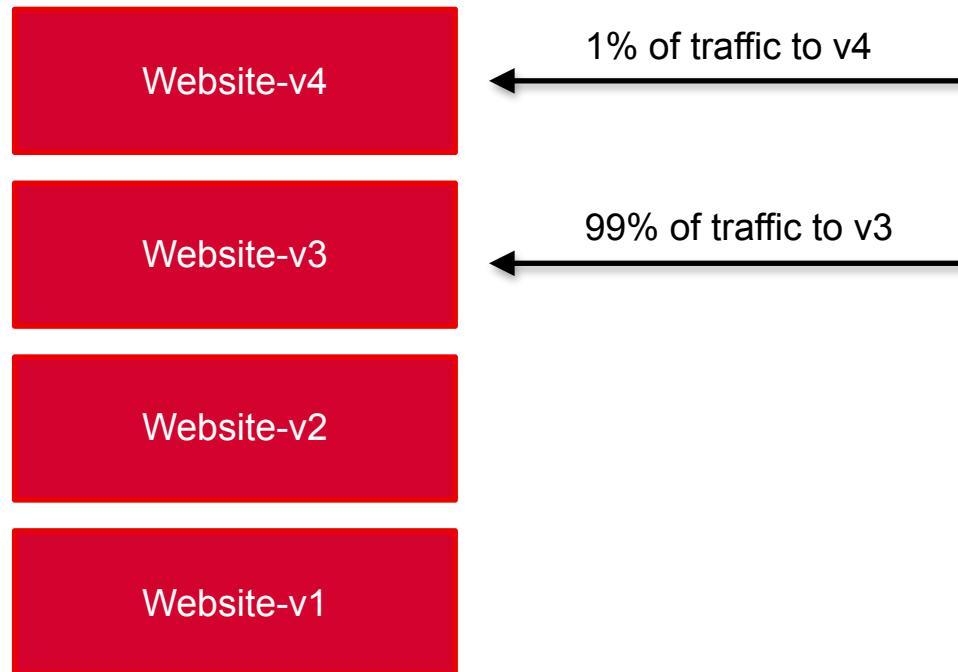
# Revisions are Immutable



Once a revision is created it is locked and cannot be modified. If you need to change anything a new revision should be created



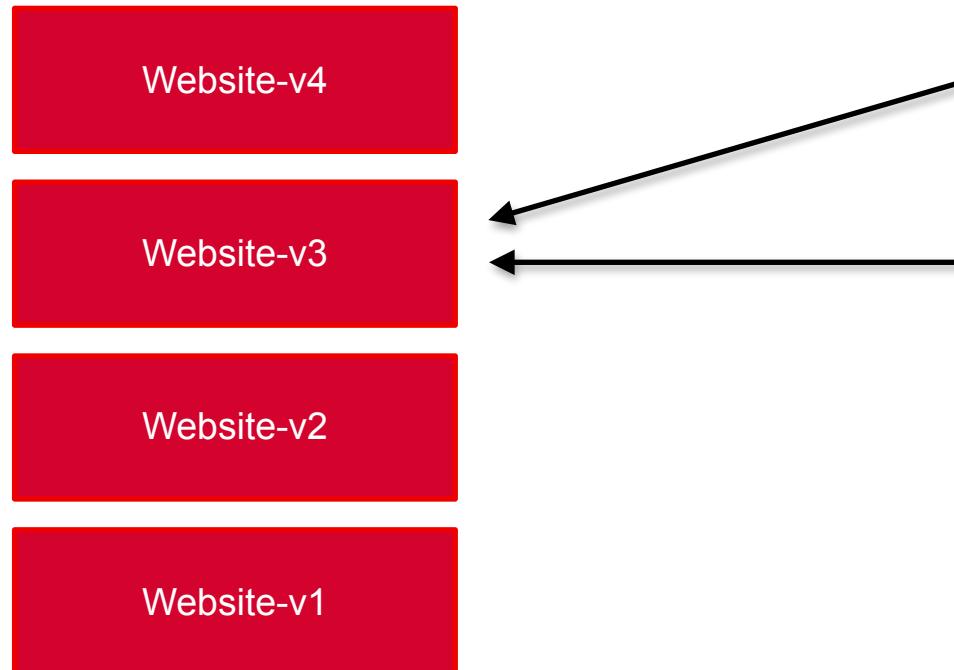
# Traffic Splitting



Allows for the gradual rollout of new features by enabling them only for a percentage of your users



# Automatic Rollback



Can redirect traffic to older versions if a new revision causes issues



# Concurrency and Scaling



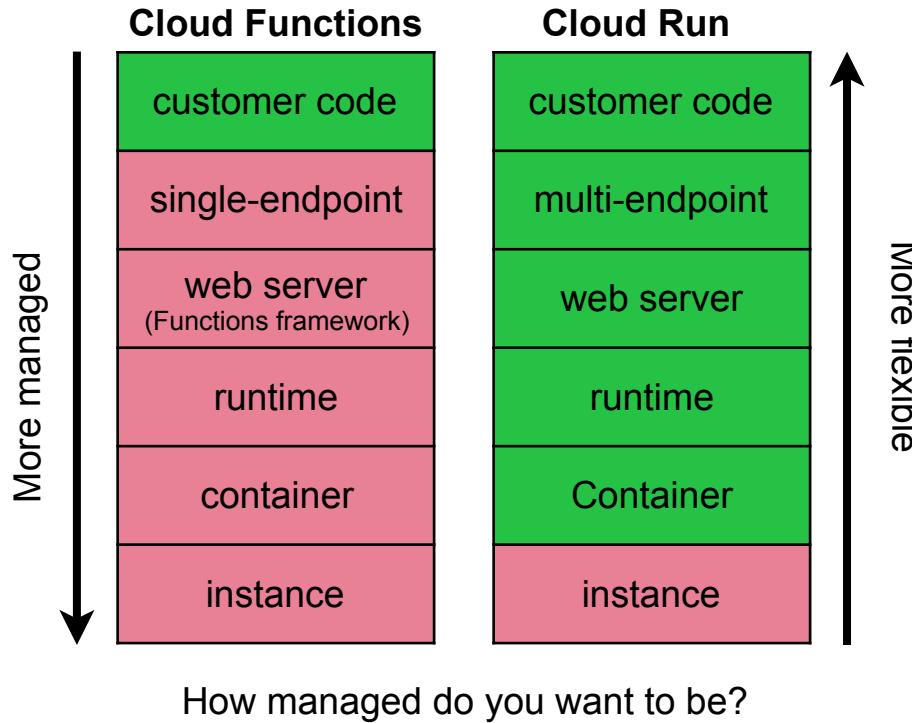
Each revision has its own concurrency and scaling settings



**Mitigating cold start – specify that a minimum number of instances of your service should be always running**



# Cloud Run Functions vs. Cloud Run





# Cloud Functions vs. Cloud Run

## Cloud Functions

- Specific limited runtimes supported
- Can be **triggered** based on platform events
- No support for running jobs
- 2nd generation functions support concurrency

## Cloud Run

- All runtimes that can be run using containers
- Expose endpoints and invoked using HTTP requests
- Support for running jobs
- Great support for concurrent requests



# Cloud Functions vs. Cloud Run

## Cloud Functions

- Choose Cloud Functions if you primarily want to connect to other cloud services on Google Cloud

## Cloud Run

- Choose Cloud Run if you want a simple way to scale and maintain services using containers



# App Engine vs. Cloud Run

## App Engine

- Porting to App Engine typically involves a **rewrite** of app
- More **restrictive** in terms of runtimes
- More managed in terms of infrastructure and scaling
- Better support for automated traffic splitting and scaling

## Cloud Run

- Porting to Cloud Run involves **containerization** of app not rewrite
- More **flexible** since you can run any runtime in a container
- More control over your containerized environment
- More set up for automated traffic splitting and scaling

# Cloud Functions and Cloud Run

Your development team has built a containerized application and already has the Docker image ready to deploy. The team wants to deploy the application on Google Cloud but doesn't have the bandwidth to manage the underlying infrastructure. They are also expecting the service to gain popularity quickly and need autoscaling capabilities to handle increased traffic without manual intervention. What should you do to meet their needs?

- A. Set up a managed instance group (MIG) and manually scale the container instances.
- B. Deploy the container on Google Kubernetes Engine (GKE) and configure cluster autoscaling.
- C. Register the Docker image with the Artifact Registry and use Cloud Run to deploy your application.
- D. Use Cloud Functions to deploy the container and handle autoscaling automatically.



# Cloud Functions and Cloud Run

Your development team has built a containerized application and already has the Docker image ready to deploy. The team wants to deploy the application on Google Cloud but doesn't have the bandwidth to manage the underlying infrastructure. They are also expecting the service to gain popularity quickly and need autoscaling capabilities to handle increased traffic without manual intervention. What should you do to meet their needs?

- A. Set up a managed instance group (MIG) and manually scale the container instances.
- B. Deploy the container on Google Kubernetes Engine (GKE) and configure cluster autoscaling.
- C. Register the Docker image with the Artifact Registry and use Cloud Run to deploy your application.**
- D. Use Cloud Functions to deploy the container and handle autoscaling automatically.



# Cloud Functions and Cloud Run

Your company has an application that uploads images to a Cloud Storage bucket, and you need to process each image as soon as it becomes available. The processing should start automatically when a new image is uploaded, and you want to minimize costs by only paying for the compute resources when the processing happens. Additionally, your team does not want to manage any infrastructure. What should you do?

- A. Deploy a managed instance group (MIG) to continuously monitor the bucket and process the images.
- B. Set up a Kubernetes cluster and configure autoscaling to process images when they are uploaded.
- C. Deploy your image processing code in a Cloud Function triggered by the Cloud Storage bucket.
- D. Use a Cloud Run service to handle image processing, scaling based on incoming requests.



# Cloud Functions and Cloud Run

Your company has an application that uploads images to a Cloud Storage bucket, and you need to process each image as soon as it becomes available. The processing should start automatically when a new image is uploaded, and you want to minimize costs by only paying for the compute resources when the processing happens. Additionally, your team does not want to manage any infrastructure. What should you do?

- A. Deploy a managed instance group (MIG) to continuously monitor the bucket and process the images.
- B. Set up a Kubernetes cluster and configure autoscaling to process images when they are uploaded.
- C. Deploy your image processing code in a Cloud Function triggered by the Cloud Storage bucket.**
- D. Use a Cloud Run service to handle image processing, scaling based on incoming requests.



# Cloud Functions and Cloud Run

Your team is deploying a new feature for a web application using Google Cloud's serverless technology, Cloud Run. To minimize risk, you want to gradually roll out the new feature to 20% of the users, while keeping 80% of the traffic on the stable version of the application. Once you're confident the new version is working well, you plan to increase traffic to it. How can you achieve this?

- A. Set up traffic splitting between Cloud Run revisions, sending 80% of traffic to the current revision and 20% to the new revision.
- B. Deploy both versions of the application using Cloud Functions and manually adjust the traffic allocation.
- C. Configure traffic splitting between Cloud Run versions, directing 80% of users to the stable version and 20% to the new one.
- D. Use Kubernetes Engine with an external load balancer to control the traffic distribution between the old and new versions.



# Cloud Functions and Cloud Run

Your team is deploying a new feature for a web application using Google Cloud's serverless technology, Cloud Run. To minimize risk, you want to gradually roll out the new feature to 20% of the users, while keeping 80% of the traffic on the stable version of the application. Once you're confident the new version is working well, you plan to increase traffic to it. How can you achieve this?

- A. Set up traffic splitting between Cloud Run revisions, sending 80% of traffic to the current revision and 20% to the new revision.**
- B. Deploy both versions of the application using Cloud Functions and manually adjust the traffic allocation.
- C. Configure traffic splitting between Cloud Run versions, directing 80% of users to the stable version and 20% to the new one.
- D. Use Kubernetes Engine with an external load balancer to control the traffic distribution between the old and new versions.



# Instance Groups



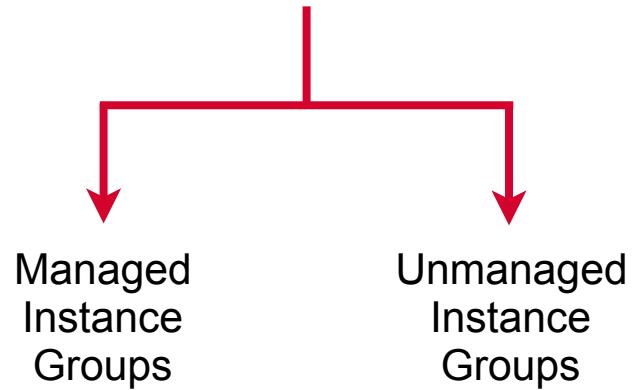


# Instance Groups

A collection of virtual machines that you can manage as a single entity

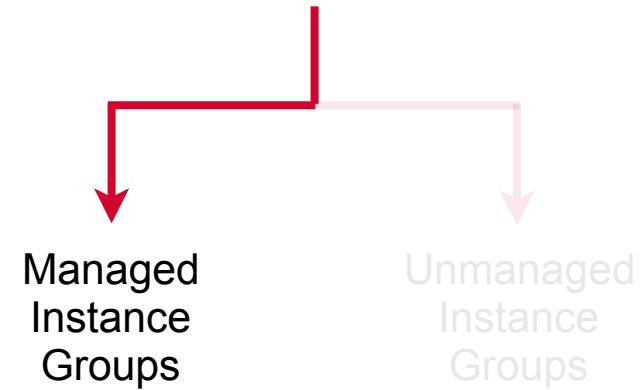


# Instance Groups





# Instance Groups





# Managed Instance Group

Group of identical GCE VM instances, created from the same instance template that are managed by the platform



# Managed Instance Group

Group of identical GCE VM instances, created from the same instance template that are managed by the platform

Instances have the exact same configuration



# Managed Instance Group

Group of identical GCE VM instances, **created from the same instance template** that are managed by the platform

The configuration is specified in  
an instance template



# Instance Template

A specification of machine type, boot disk (or container image), zone, labels and other instance properties that can be used to instantiate either individual VM instances or a Managed Instance Group



# Instance Template to Create Instances



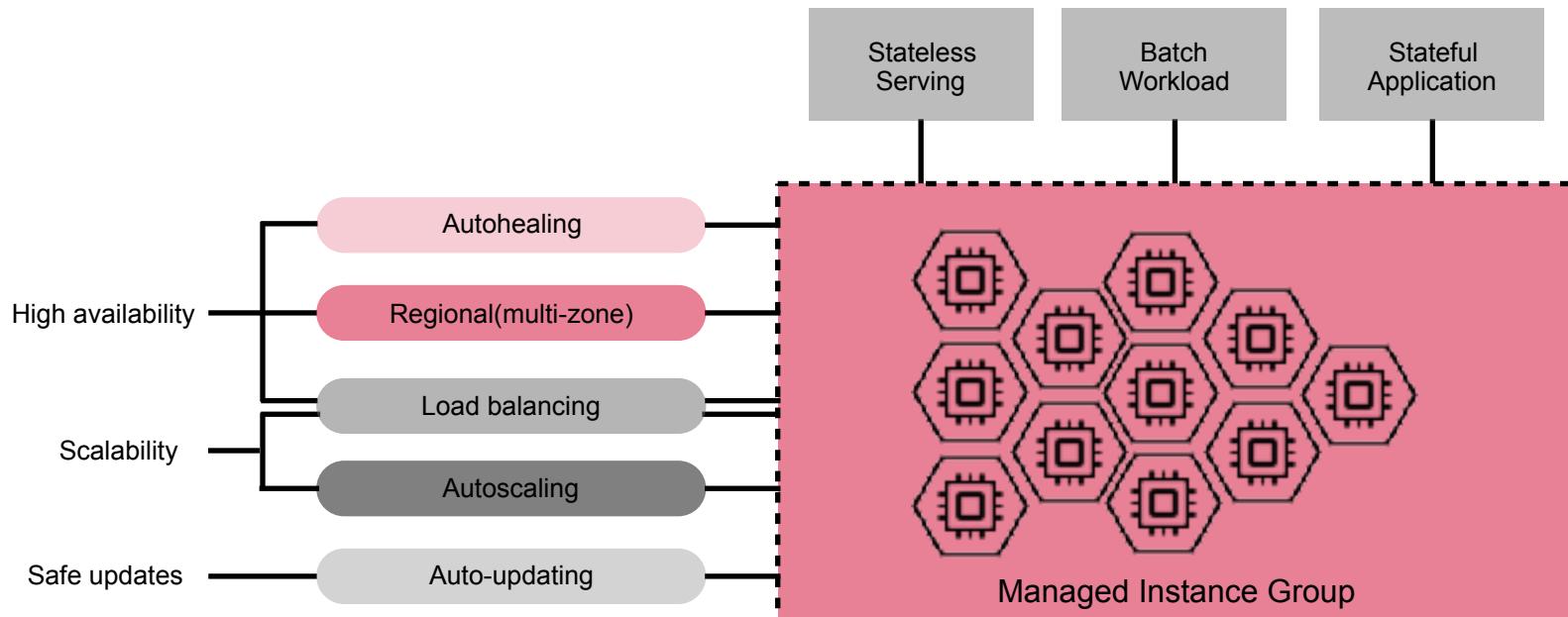
Instance Template



Managed Instance  
Group

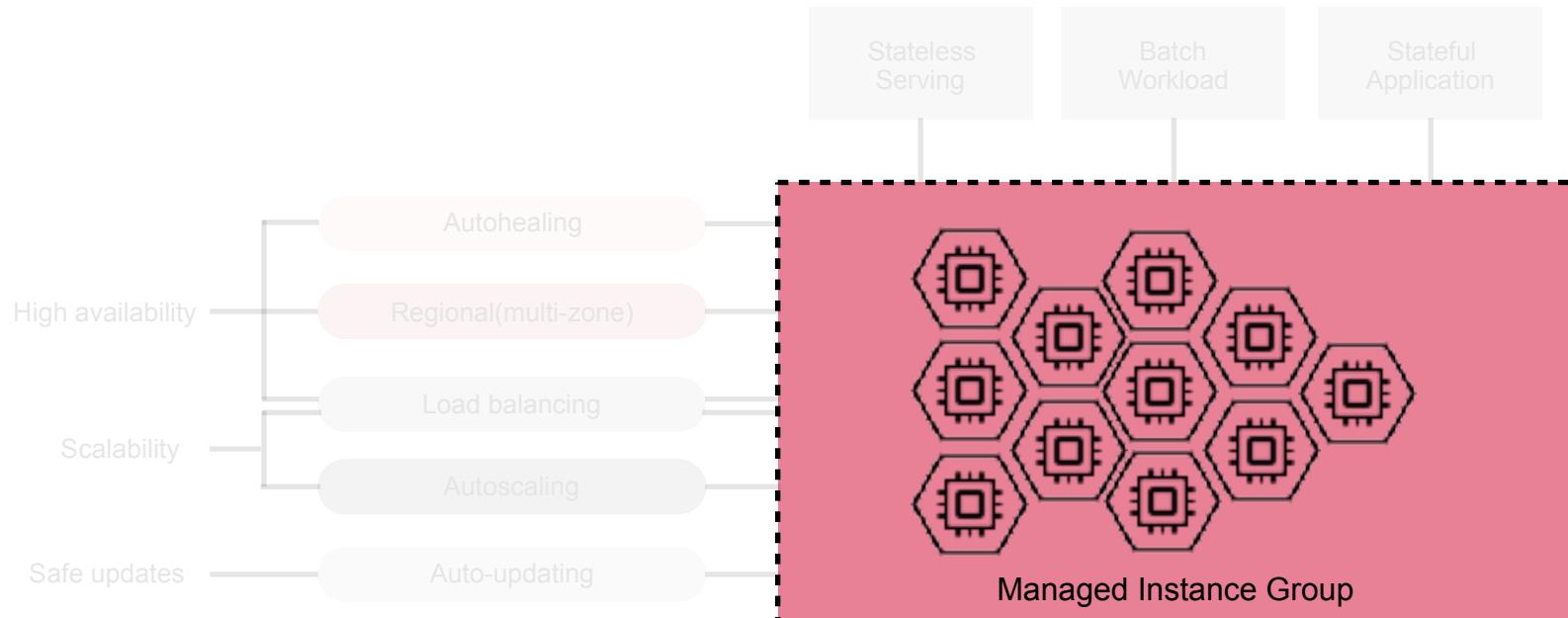


# Managed Instance Groups



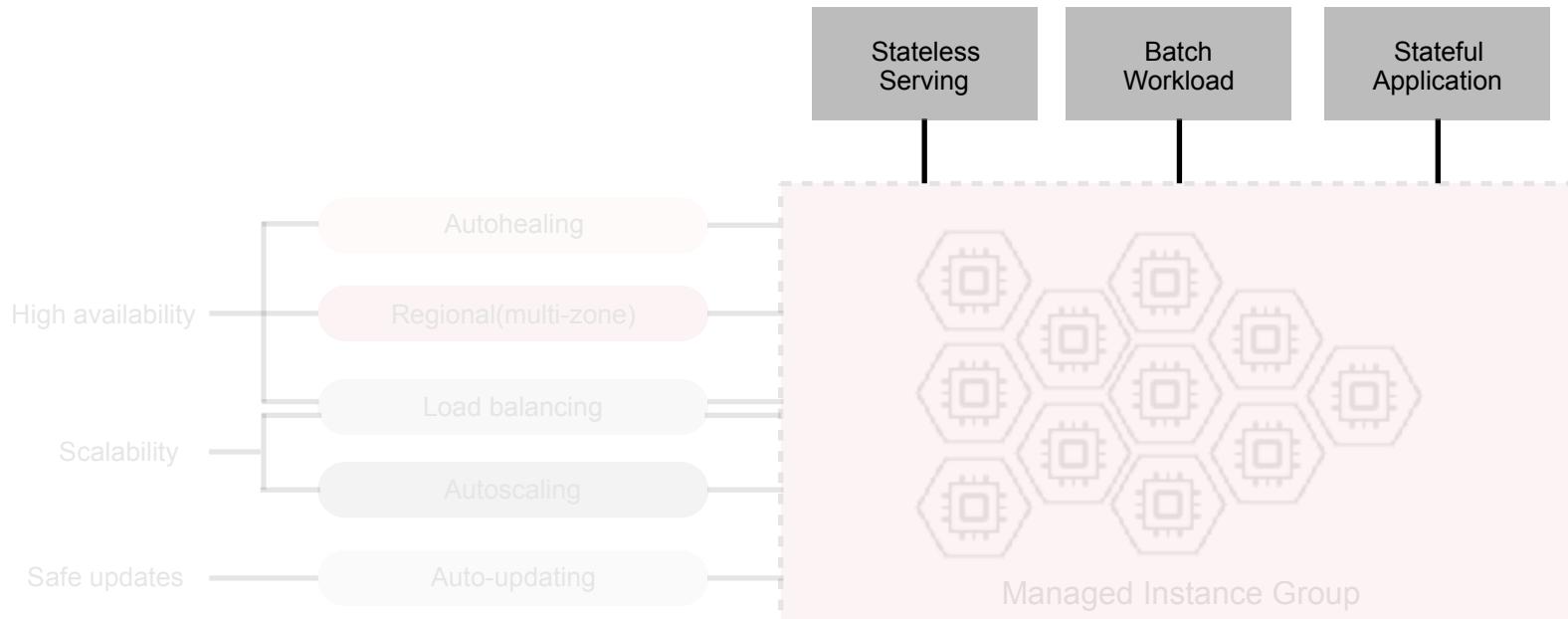


# Multiple Instances Created from the Same Template





# Managed Instance Groups





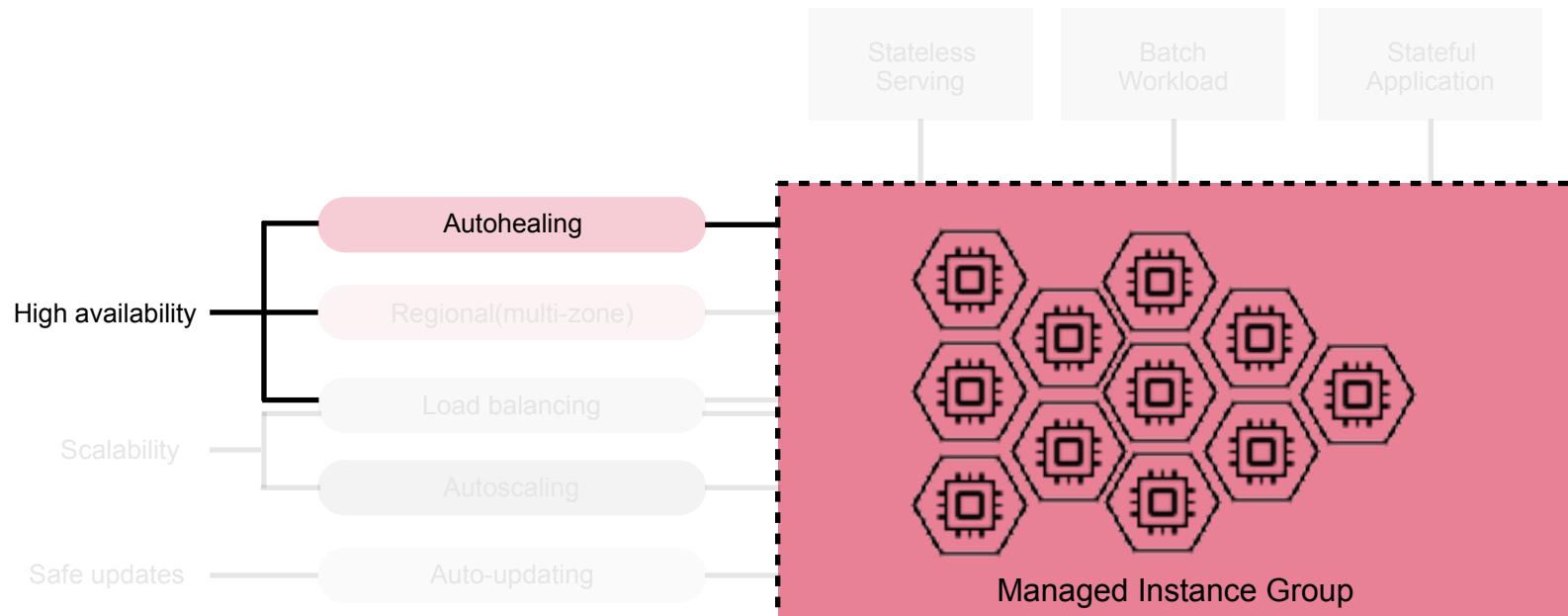
# Choosing Managed Instance Groups

- Stateless serving workloads,
  - e.g. a website frontend
- Stateless batch, high-performance, high throughput workloads
  - e.g. image processing from a queue
- Stateful applications
  - e.g. databases, legacy applications, long-running batch computations
  - MIG will preserve each instance's unique state such as persistent disks, metadata



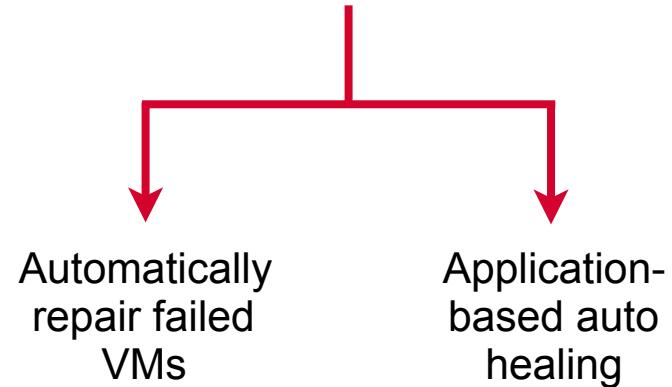


# Autohealing



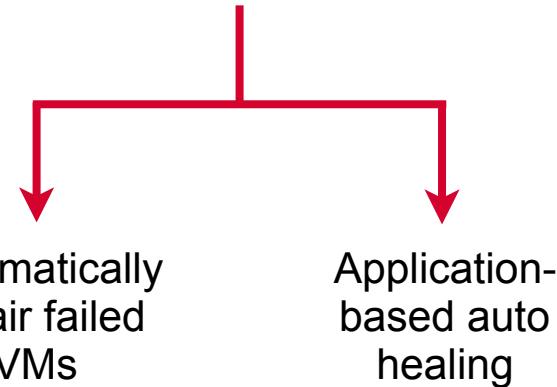


# Autohealing





# Autohealing

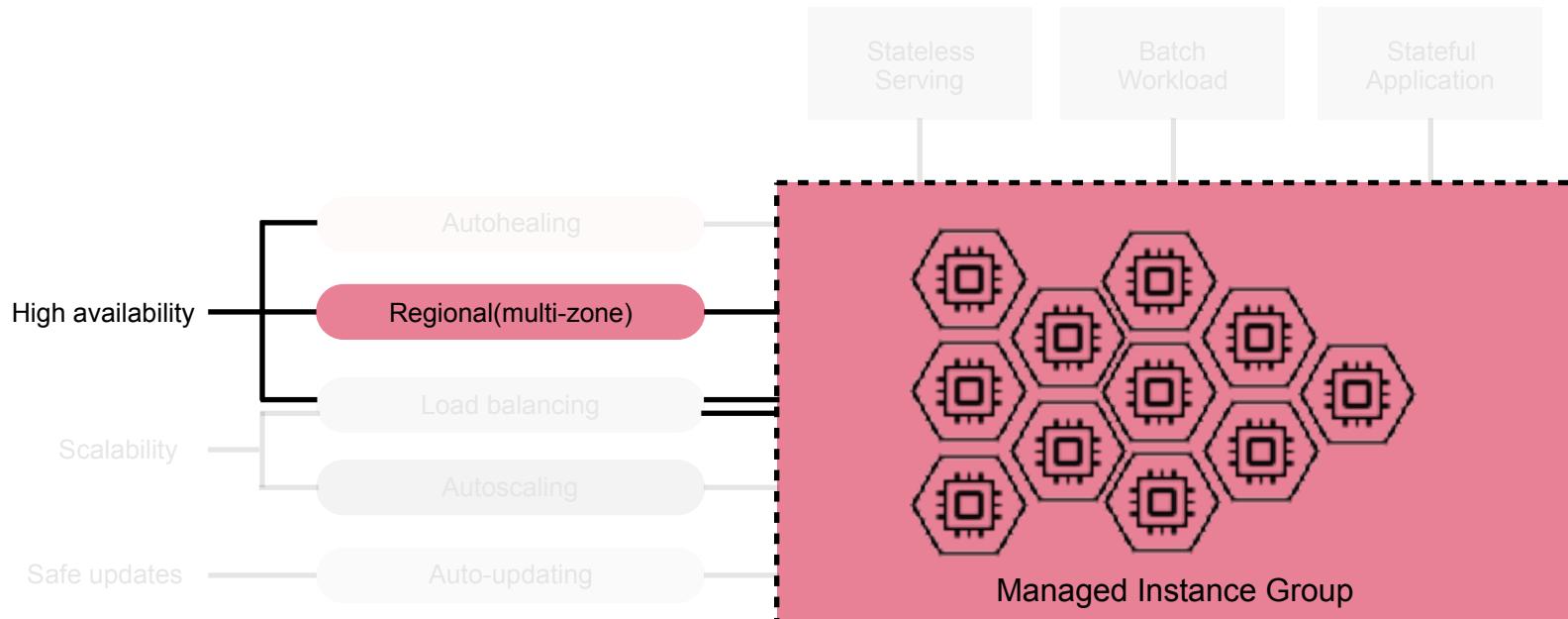


If VM stops, crashes, or is pre-empted (spot VMs) it is automatically recreated based on the template

Application-based health checks to check whether the application is responding as expected. If not the VM is recreated



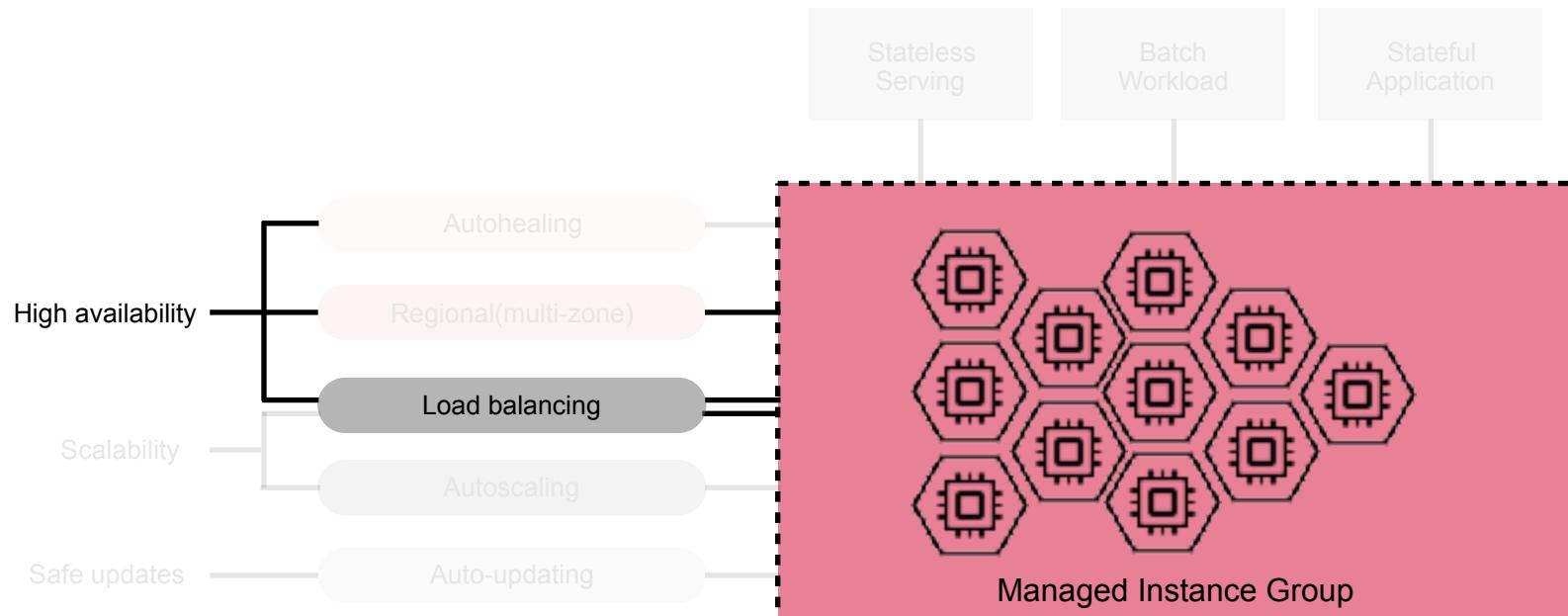
# Regional (Multi-zone) Clusters



This replication protects against zonal failures. If that happens, your app can continue serving traffic from instances running in the remaining available zones in the same region.



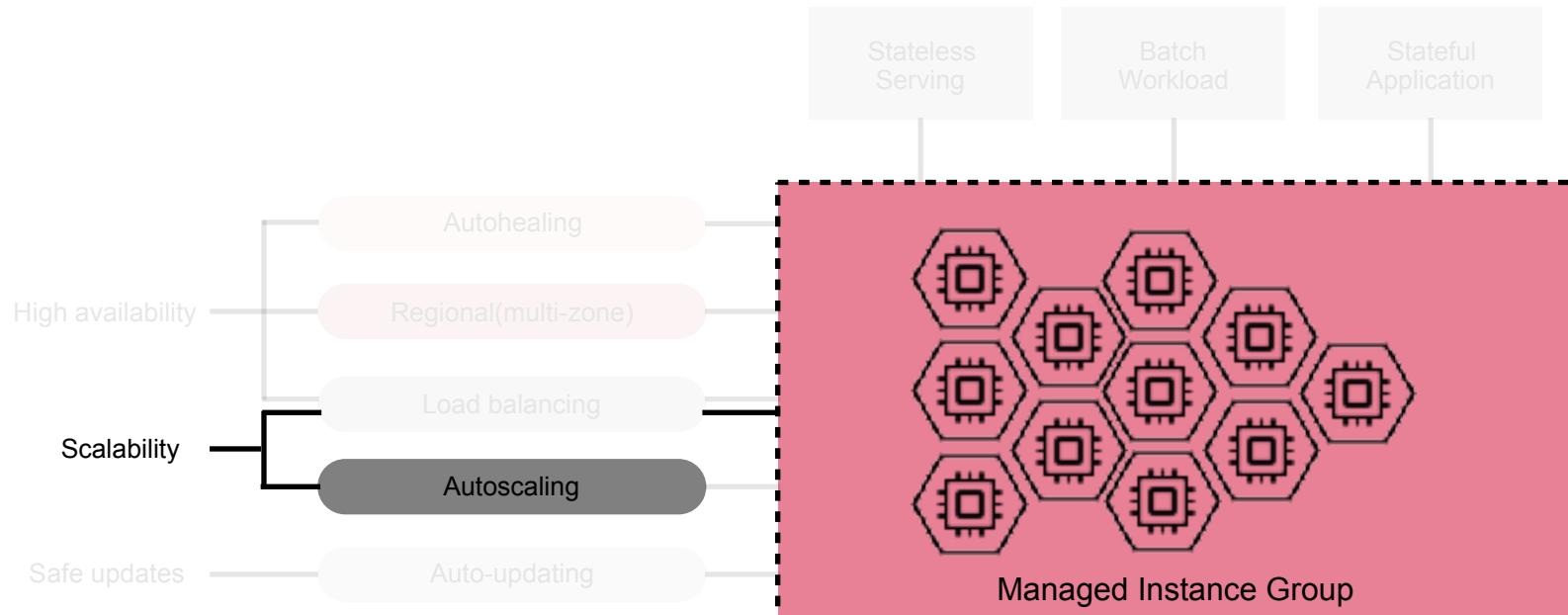
# Load Balancing



Can work with load balancing services to distribute traffic across the instances of the group



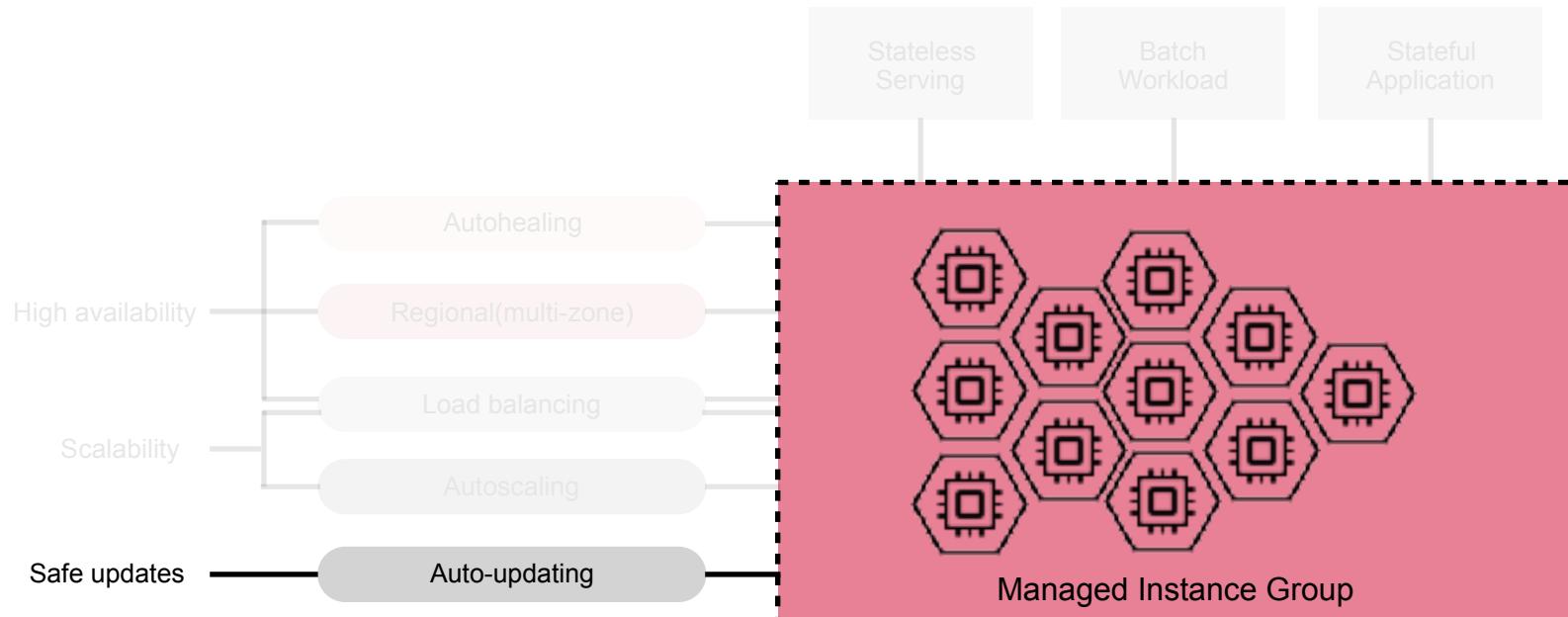
# Scalability



The number of instances can grow automatically to meet demand and will shrink automatically when demand drops



# Scalability



Can deploy new versions of software to instances in your MIG -  
supports **rolling updates and canary updates (partial rollouts)**



# VMs for Other Workloads

GPU VMs

Groups of Pre-emptible instances

VMs running Containers



# VMs for Other Workloads

GPU VMs

Groups of Pre-emptible instances

VMs running Containers

Run workloads that need GPUs - configure the specification in the instance template



# VMs for Other Workloads

GPU VMs

Groups of Pre-emptible instances

VMs running Containers

Reduce cost of workloads by using pre-emptible instances. When instances are pre-empted the MIG will recreate them when capacity is available



# VMs for Other Workloads

GPU VMs

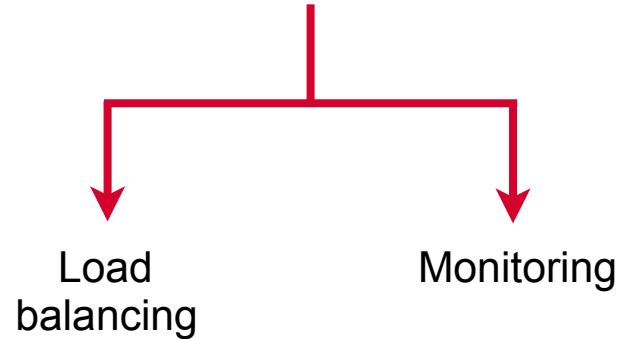
Groups of Pre-emptible instances

VMs running Containers

Can specify container images in the instance template. Use the container-optimized OS and the container will be started on each VM

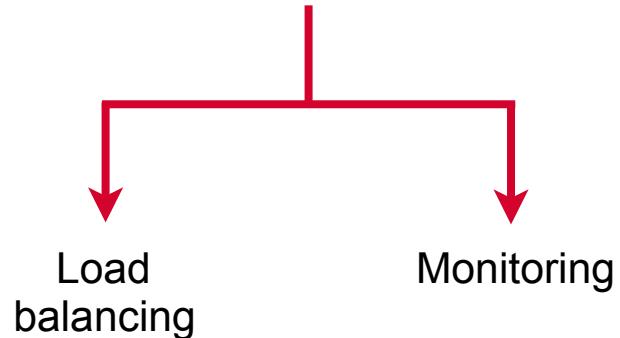


# Health Checks





# Health Checks



Direct traffic from non-responsive instances towards healthy instances

Monitoring

Signal to the MIG to proactively delete and recreate instances that become unhealthy



**Use separate health checks for load balancing and monitoring - do not rely on the same health check**



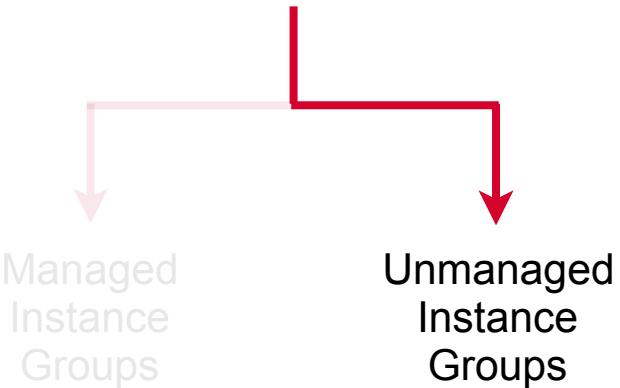
# Some Notes on MIGs

- Instance templates **cannot be edited** - create a **new instance template** and associate that with the group
- **Configuration errors in the instance template** can cause instance creation to fail
- If there are **existing instances or disks with the same name**, instance creation can fail
  - Good practice to configure auto-delete on persistent disks attached to instances





# Instance Groups



Unmanaged instance groups can contain heterogeneous instances that you can arbitrarily add and remove from the group.

Do not offer autoscaling, auto healing - can be used with a load balancer

# Managed Instance Groups

Your company runs a critical application using a managed instance group (MIG) on Google Cloud. Recently, you've found that instance creation is failing. The new instances need to have updated configurations that require different storage setups. What should you do to resolve the issue and ensure the instances are created successfully?

- A. Manually rename the persistent disks in the existing instances to avoid conflicts.
- B. Create a new instance template with the right configuration and delete existing persistent disks with the same name as new instances.
- C. Update the MIG's autoscaler to force the creation of new instances despite the conflicts using the —force option
- D. Modify the existing instance template and ignore any disk-related errors during instance creation and rely on the autoscaler to fix this



# Managed Instance Groups

Your company runs a critical application using a managed instance group (MIG) on Google Cloud. Recently, you've found that instance creation is failing. The new instances need to have updated configurations that require different storage setups. What should you do to resolve the issue and ensure the instances are created successfully?

- A. Manually rename the persistent disks in the existing instances to avoid conflicts.
- B. Create a new instance template with the right configuration and delete existing persistent disks with the same name as new instances.**
- C. Update the MIG's autoscaler to force the creation of new instances despite the conflicts using the —force option
- D. Modify the existing instance template and ignore any disk-related errors during instance creation and rely on the autoscaler to fix this



# Managed Instance Groups

Your company is migrating a legacy application to Google Cloud. The application connects to a database and requires each instance to maintain session data and preserve state across multiple requests. You want to take advantage of autoscaling to handle fluctuating traffic. What should you do to meet these requirements?

- A. Use a regular unmanaged instance group and manually configure autoscaling.
- B. Set up a stateless managed instance group with an external load balancer.
- C. Create a managed instance group for stateful applications to preserve instance state
- D. Configure a preemptible instance group for cost-effective scaling of the application.



# Managed Instance Groups

Your company is migrating a legacy application to Google Cloud. The application connects to a database and requires each instance to maintain session data and preserve state across multiple requests. You want to take advantage of autoscaling to handle fluctuating traffic. What should you do to meet these requirements?

- A. Use a regular unmanaged instance group and manually configure autoscaling.
- B. Set up a stateless managed instance group with an external load balancer.
- C. Create a managed instance group for stateful applications to preserve instance state**
- D. Configure a preemptible instance group for cost-effective scaling of the application.



# Managed Instance Groups

Your company is using a managed instance group (MIG) on Google Cloud to automatically scale an application based on traffic demand. Recently, you noticed that the MIG is creating more instances than necessary during autoscaling. Each instance takes about 2 minutes to start up and become fully operational. The health check is currently configured to run every 30 seconds. What should you do to resolve this issue?

- A. Decrease the health check interval to 15 seconds to detect healthy instances faster.
- B. Increase the cooldown period of the MIG to allow more time between instance launches.
- C. Configure the health check to run every 2.5 minutes so the instances have time to start up and become responsive.
- D. Modify the autoscaling policy to create fewer instances at a time to avoid over-provisioning.



# Managed Instance Groups

Your company is using a managed instance group (MIG) on Google Cloud to automatically scale an application based on traffic demand. Recently, you noticed that the MIG is creating more instances than necessary during autoscaling. Each instance takes about 2 minutes to start up and become fully operational. The health check is currently configured to run every 30 seconds. What should you do to resolve this issue?

- A. Decrease the health check interval to 15 seconds to detect healthy instances faster.
- B. Increase the cooldown period of the MIG to allow more time between instance launches.
- C. Configure the health check to run every 2.5 minutes so the instances have time to start up and become responsive.**
- D. Modify the autoscaling policy to create fewer instances at a time to avoid over-provisioning.



# Google Cloud: Associate Cloud Engineer Bootcamp - Day 2





## Day 2: Course Schedule

- Taxonomy of Storage Solutions
- Google Cloud Storage
- Containers and Kubernetes
- Load Balancing
- Identity and Access Management
- IAM Best Practices
- Organization Policy Service
- Billing Accounts
- Batch Processing vs. Stream Processing
- Pub/Sub

# Taxonomy of Storage Solutions





# Choices in Computing



**Compute**

Where is code executed and how?



**Storage**

Where is data stored?

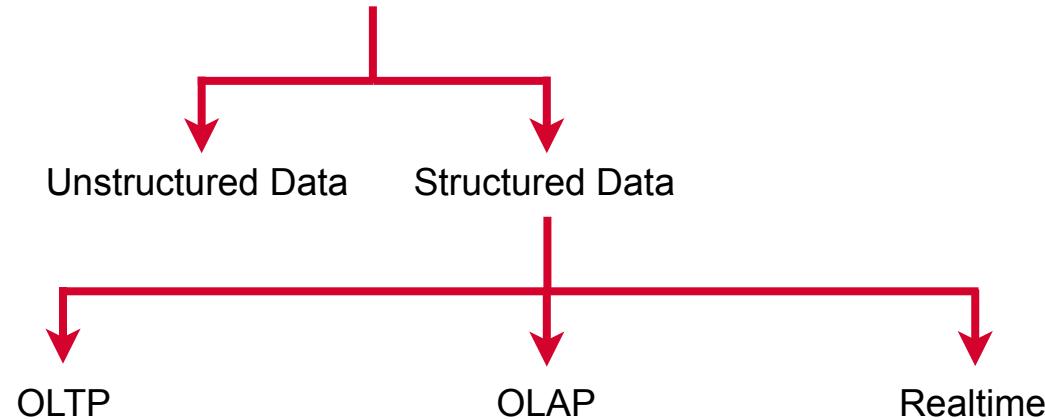


# Storage Technologies



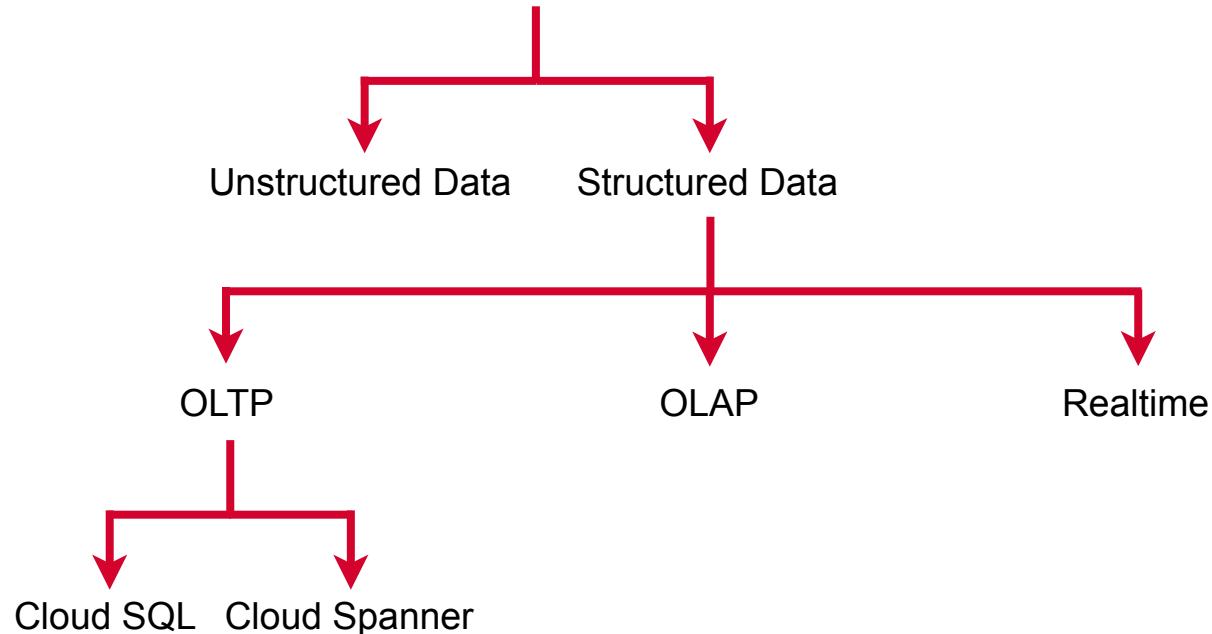


# Storage Technologies



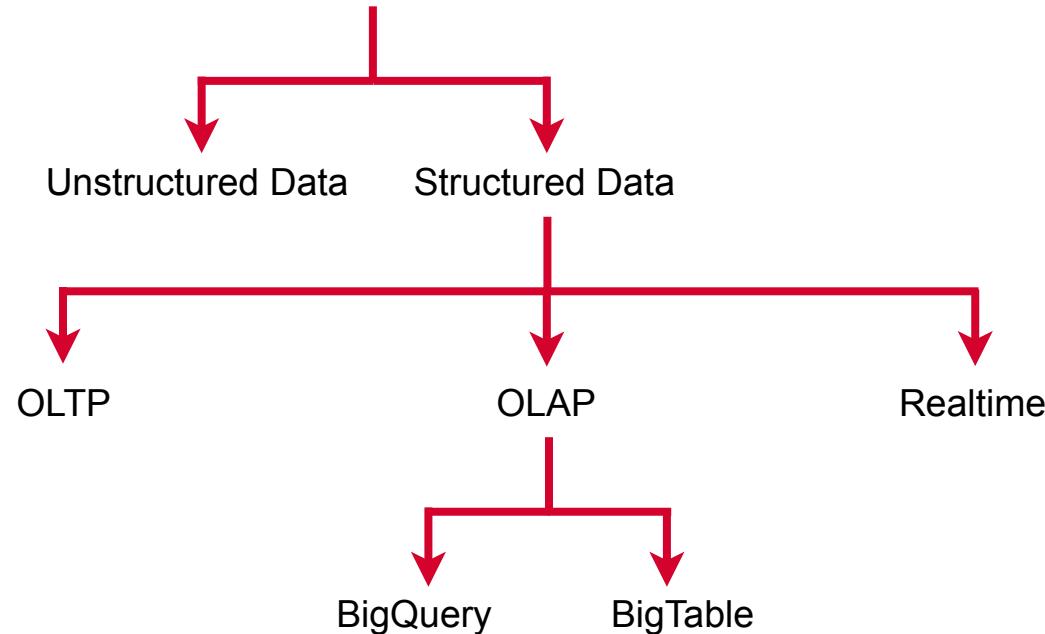


# Storage Technologies



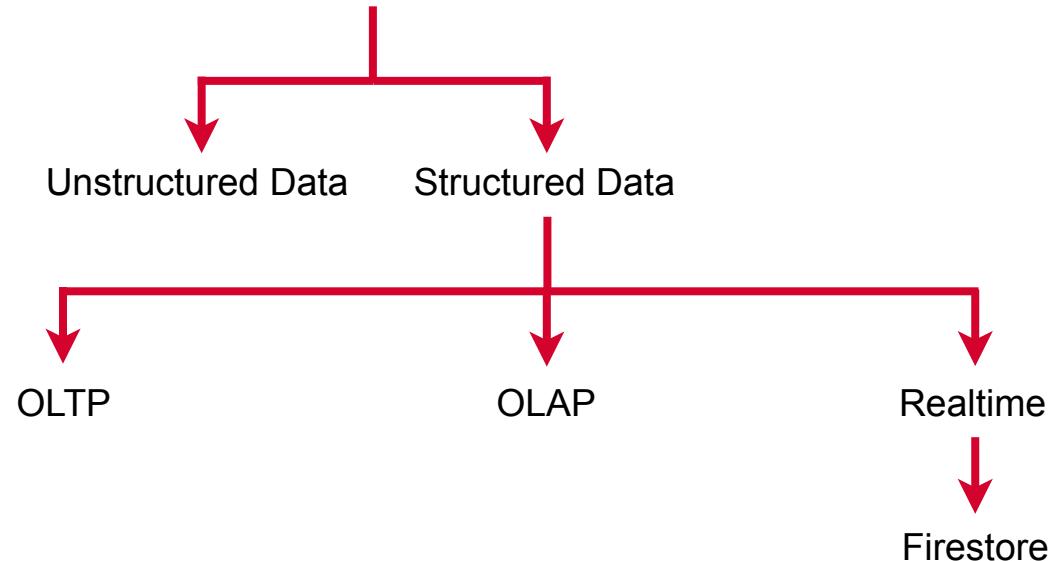


# Storage Technologies



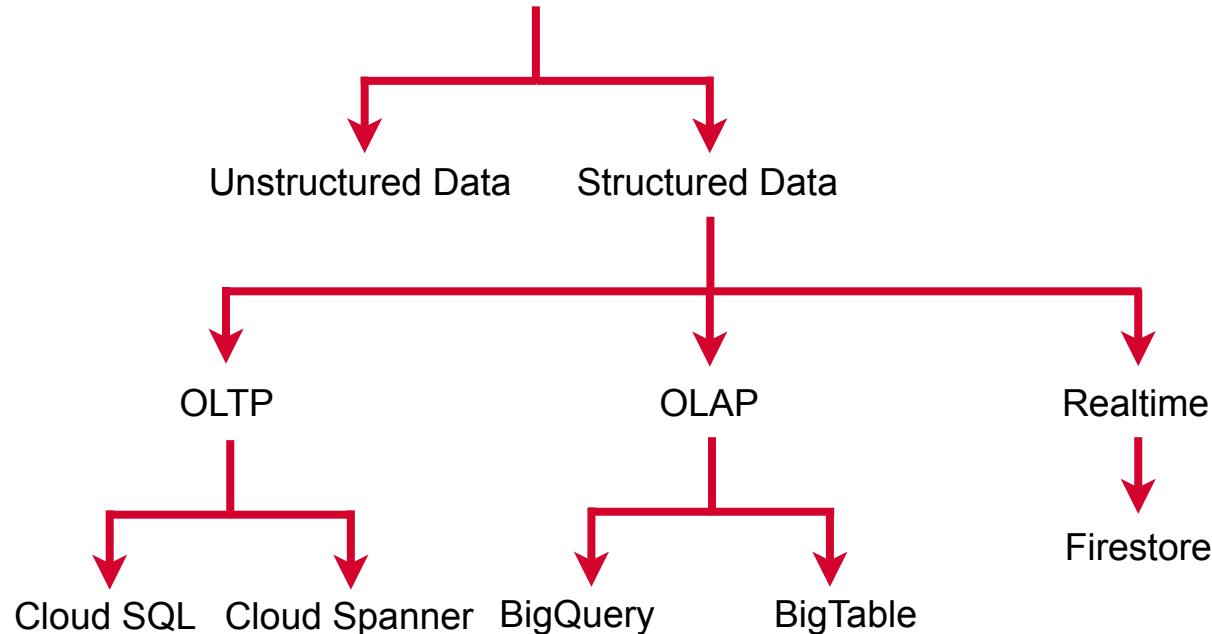


# Storage Technologies



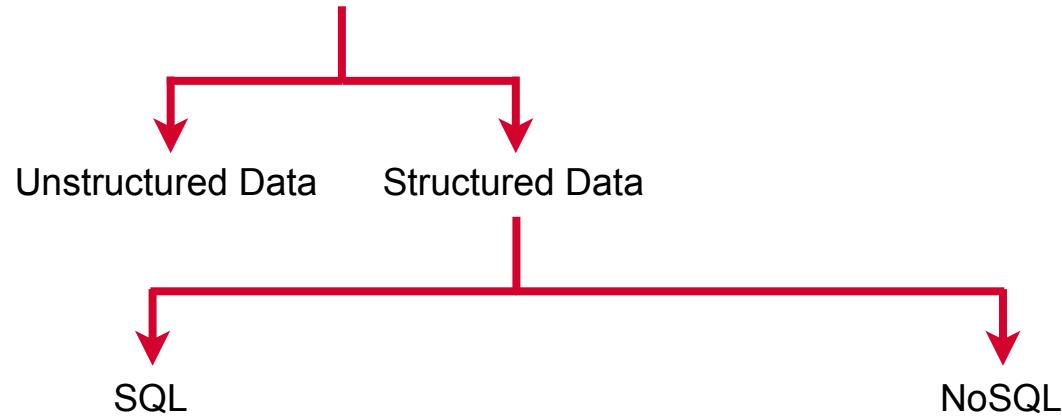


# Storage Technologies



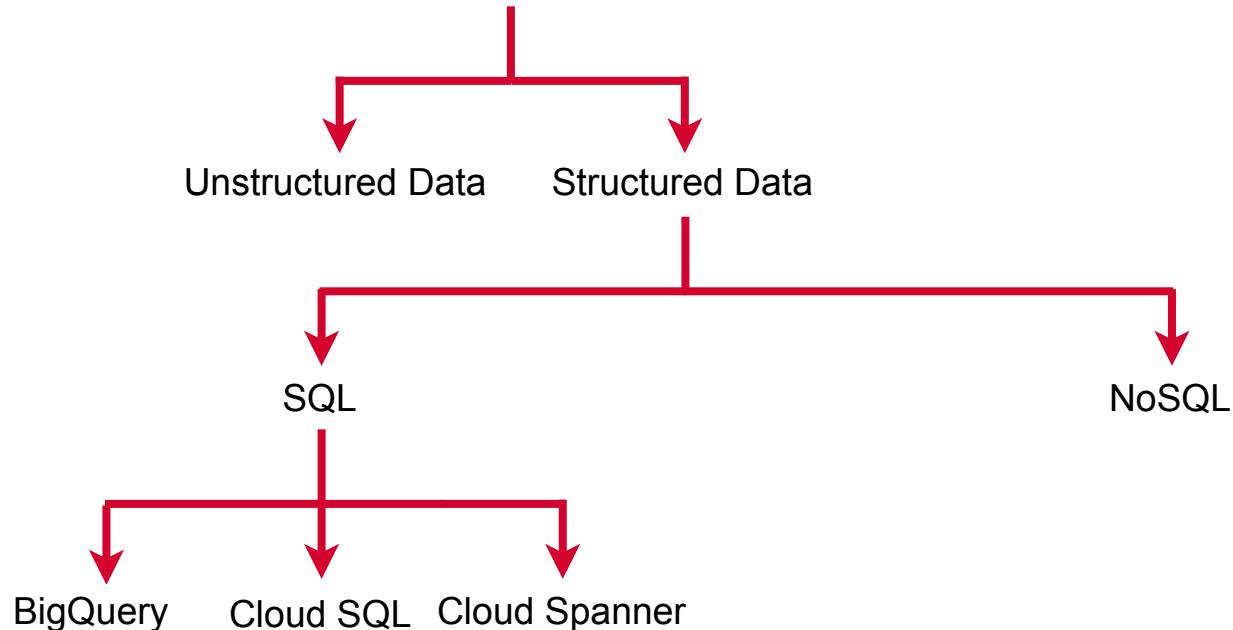


# Storage Technologies



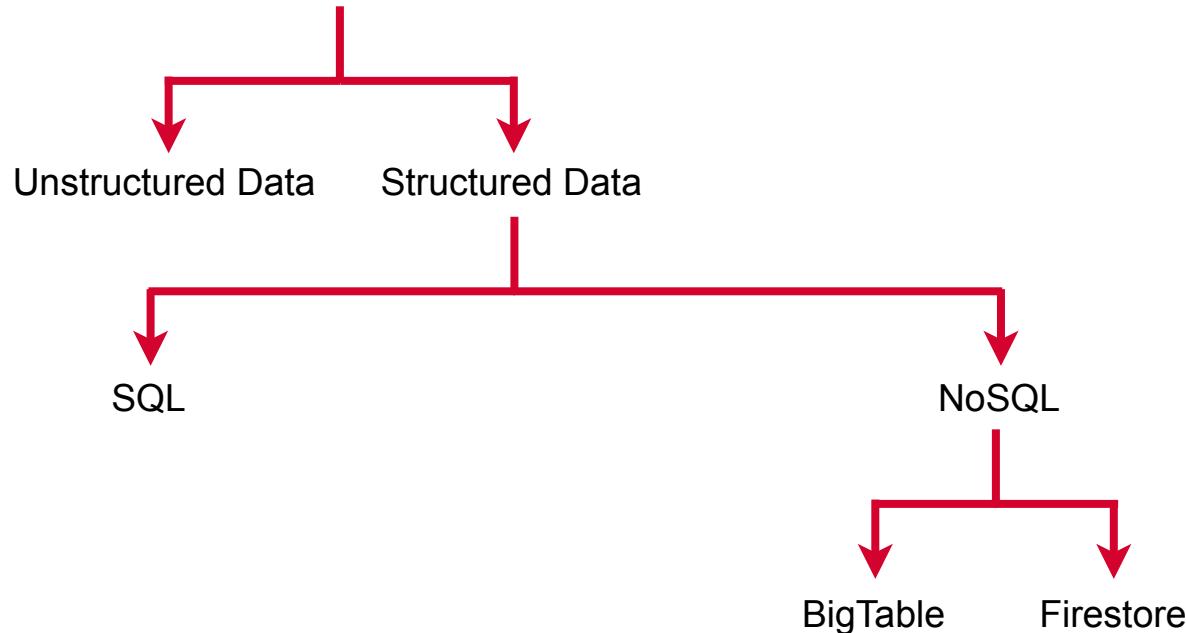


# Storage Technologies



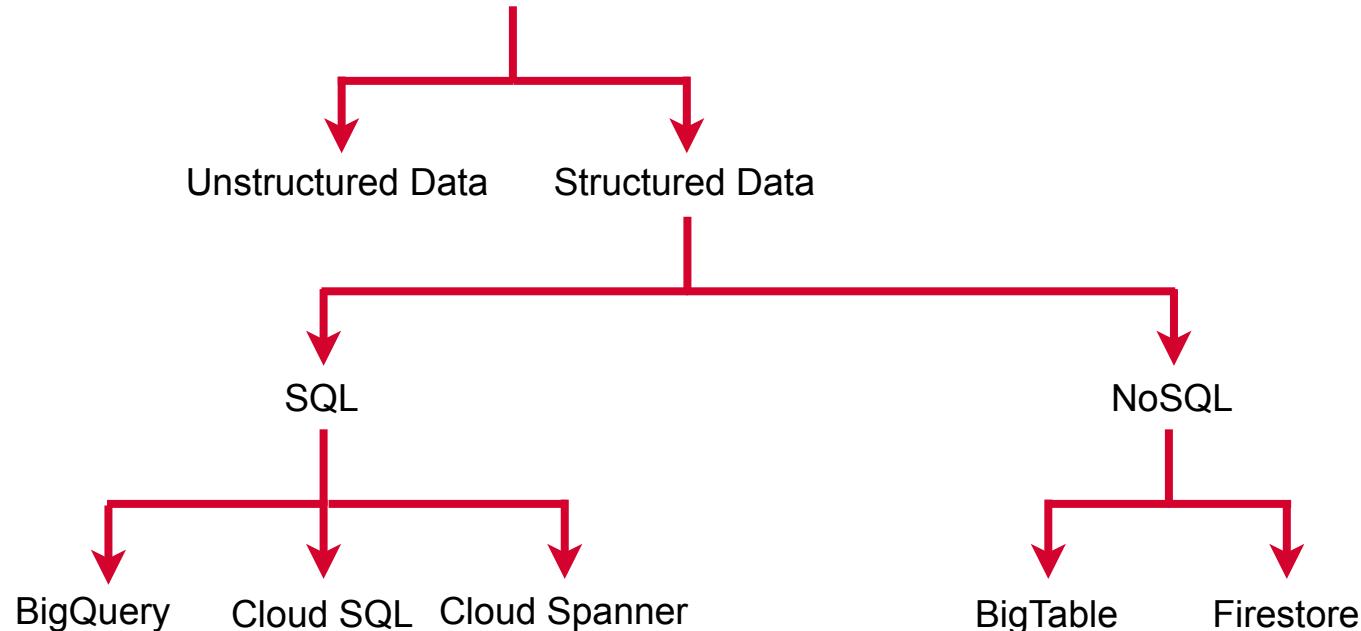


# Storage Technologies



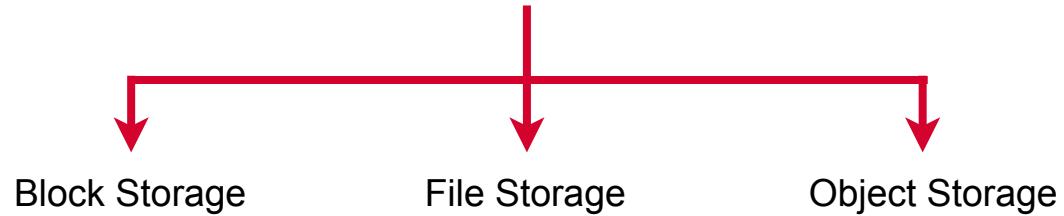


# Storage Technologies



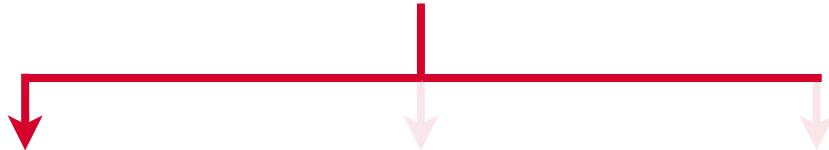


# Unstructured Data





# Unstructured Data

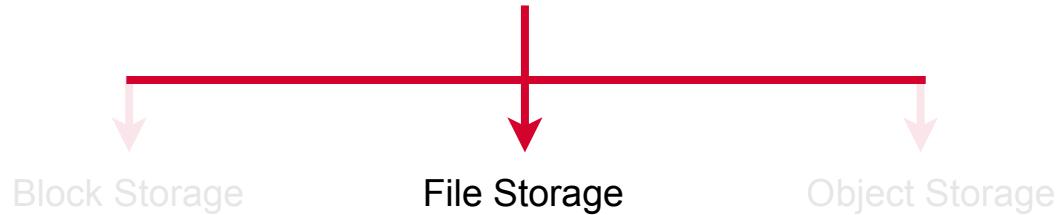


Physically addressable storage  
accessed from compute - data  
split into uniform blocks

High performance read and write  
access at the block level



# Unstructured Data

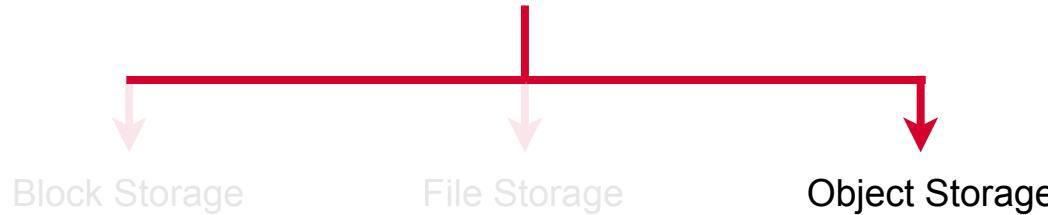


Stores data as a hierarchy of files  
within directories

Shared concurrent access from  
multiple machines



# Unstructured Data



Logically addressable  
storage accessed from  
compute or by human users



# Persistent Disks vs. Buckets

## Persistent Disks

- Block storage
- Max 64TB in size
- **Pay what you allocate**
- Tied to GCE VMs
- Zonal (or regional) access

## Buckets

- Object storage
- Infinitely scalable
- **Pay what you use**
- Independent of GCE VMs
- Global access



# Storage Use Cases

Use Case	Appropriate GCP Service	Non-GCP Equivalents
Block storage	Persistent disks or local SSDs	AWS EBS, Azure Disk
Object/blob storage	Cloud Storage (GCS) buckets	AWS S3, Azure Blob Storage
Relational data - small, regional payloads	Cloud SQL	AWS RDS, Azure SQL Database
Relational data - large, global payloads	Cloud Spanner	Aurora DB
HTML/XML documents with NoSQL access	Firestore	AWS DynamoDB, Azure Cosmos DB
Large, naturally ordered data with NoSQL access	BigTable	
Analytics and complex queries with SQL access	BigQuery	AWS Redshift, Azure Synapse Analytics



# Cloud SQL

Cloud SQL is the fully-managed MySQL, PostgreSQL and SQL Server database service on the Google Cloud Platform

Transactional support, ACID support

Easiest migration path for on-prem RDBMS

High availability using failover replicas in different zones



# Google Cloud Spanner

A **global, horizontally scaling**, strongly consistent relational database service built on proprietary technology

Scales horizontally by adding nodes

ACID support at scale

Relatively expensive and Google proprietary



# Cloud Firestore

Flexible, scalable, NoSQL database for keeping data in sync across client apps.

Mobile and web server development as a part of GCP's [Firebase](#) platform

[Realtime listeners and offline support](#)



# BigQuery Features

- **Serverless:** No cluster, no provisioning
- Structured data with fields
- **Can ingest streaming data at scale**
- Autoscaling
- Automatic high availability
- Simple SQL queries





# BigQuery Data Transfer Service

- Automates ingestion of analytics data from various sources into BigQuery
- Allows for transfers from:
  - Google services (Ads, YT)
  - External cloud services (Redshift)
- No need to write custom code to perform the transfer





# Redis

Very popular in-memory key-value NoSQL database



# Memcached

General purpose, distributed, memory-caching system



# Cloud Memorystore

Google managed service for Redis and Memcached that offers scaling, high availability and a convenient migration path



# Google Cloud Bigtable

NoSQL database technology ideal for very large, sparse datasets with sequential ordering in key column; provides very fast writes as well as reads



# Choose Bigtable For

- **Time series data:** Naturally ordered
- **Internet of Things data:** Constant stream of writes
- **Financial data:** Often efficiently represented as time series data
- **Large datasets** > 1 TB with each row < 10 MB



# Storage Solutions

You are designing a backend service for an e-commerce platform that will handle and store transactional data from mobile and web users located worldwide. With the platform's global launch, you expect to handle petabytes of transaction data, and the business team needs the ability to run SQL queries for analysis. You need to create a data store that is highly available, scalable, and capable of efficiently handling massive volumes of data. What should you do?

- A. Use Cloud Spanner to create a globally distributed, scalable, and highly available relational database that supports SQL queries.
- B. Use Firestore in Datastore mode to store transactional data and run SQL queries through third-party tools.
- C. Set up a Cloud SQL instance with regional replication to handle the large volume of global transactions.
- D. Deploy a MySQL database on Compute Engine VMs and configure replication across regions to handle the data load.



# Storage Solutions

You are designing a backend service for an e-commerce platform that will handle and store transactional data from mobile and web users located worldwide. With the platform's global launch, you expect to handle petabytes of transaction data, and the business team needs the ability to run SQL queries for analysis. You need to create a data store that is highly available, scalable, and capable of efficiently handling massive volumes of data. What should you do?

- A. **Use Cloud Spanner to create a globally distributed, scalable, and highly available relational database that supports SQL queries.**
- B. Use Firestore in Datastore mode to store transactional data and run SQL queries through third-party tools.
- C. Set up a Cloud SQL instance with regional replication to handle the large volume of global transactions.
- D. Deploy a MySQL database on Compute Engine VMs and configure replication across regions to handle the data load.



# Storage Solutions

You are designing a system to ingest and store data from millions of IoT sensors around the world. The data will be sent continuously at high throughput, and the business team needs to query and analyze this data based on the time it was generated. You want to ensure that the system can scale efficiently and handle real-time queries. What should you do?

- A. Ingest the data into BigQuery, and partition the tables by timestamp for querying.
- B. Ingest the data into Cloud SQL and index it by timestamp for efficient queries.
- C. Ingest the data into Bigtable, and create a row key based on the event timestamp for optimized querying and scaling.
- D. Ingest the data into Firestore, and organize the data based on event timestamps for real-time analysis.



# Storage Solutions

You are designing a system to ingest and store data from millions of IoT sensors around the world. The data will be sent continuously at high throughput, and the business team needs to query and analyze this data based on the time it was generated. You want to ensure that the system can scale efficiently and handle real-time queries. What should you do?

- A. Ingest the data into BigQuery, and partition the tables by timestamp for querying.
- B. Ingest the data into Cloud SQL and index it by timestamp for efficient queries.
- C. Ingest the data into Bigtable, and create a row key based on the event timestamp for optimized querying and scaling.**
- D. Ingest the data into Firestore, and organize the data based on event timestamps for real-time analysis.



# Storage Solutions

Your company is deploying a business-critical application that requires a relational database on Google Cloud. The application must be highly available and resilient to zonal failures. You need to ensure that the database remains operational even if an entire zone becomes unavailable. What should you do to meet these requirements?

- A. Use a single Cloud SQL instance and configure automated backups for disaster recovery.
- B. Use Cloud SQL configured for high availability with failover replicas in different zones.
- C. Deploy a Cloud SQL instance in a single zone and set up replication using Cloud Spanner.
- D. Use Bigtable with replication in the same zone to ensure availability and performance.



# Storage Solutions

Your company is deploying a business-critical application that requires a relational database on Google Cloud. The application must be highly available and resilient to zonal failures. You need to ensure that the database remains operational even if an entire zone becomes unavailable. What should you do to meet these requirements?

- A. Use a single Cloud SQL instance and configure automated backups for disaster recovery.
- B. Use Cloud SQL configured for high availability with failover replicas in different zones.**
- C. Deploy a Cloud SQL instance in a single zone and set up replication using Cloud Spanner.
- D. Use Bigtable with replication in the same zone to ensure availability and performance.



# Google Cloud Storage





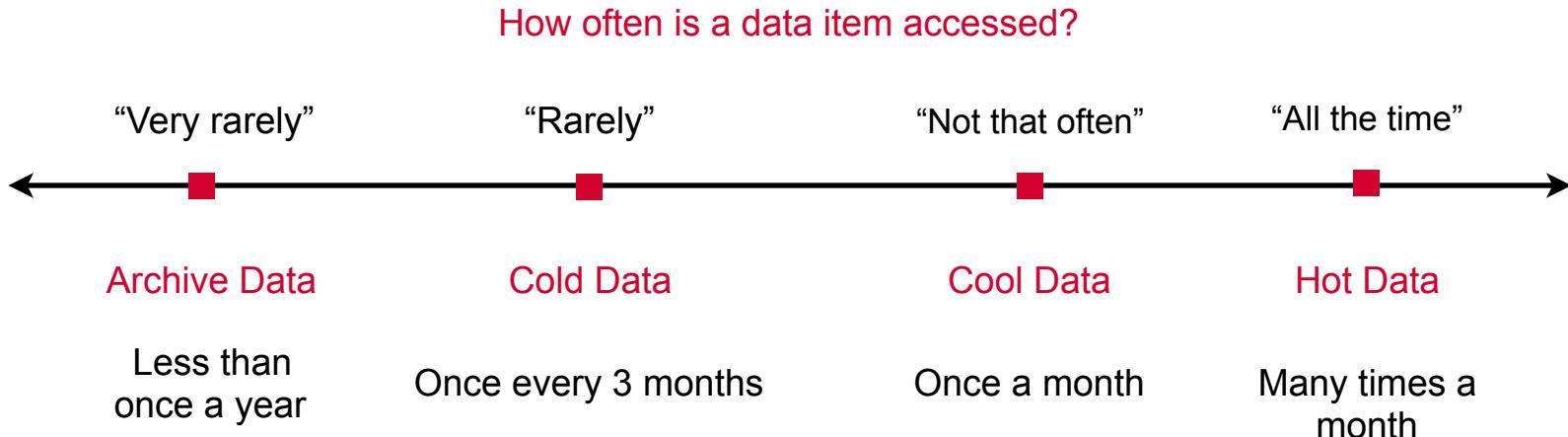
# GCS Storage Classes

How often is a data item accessed?



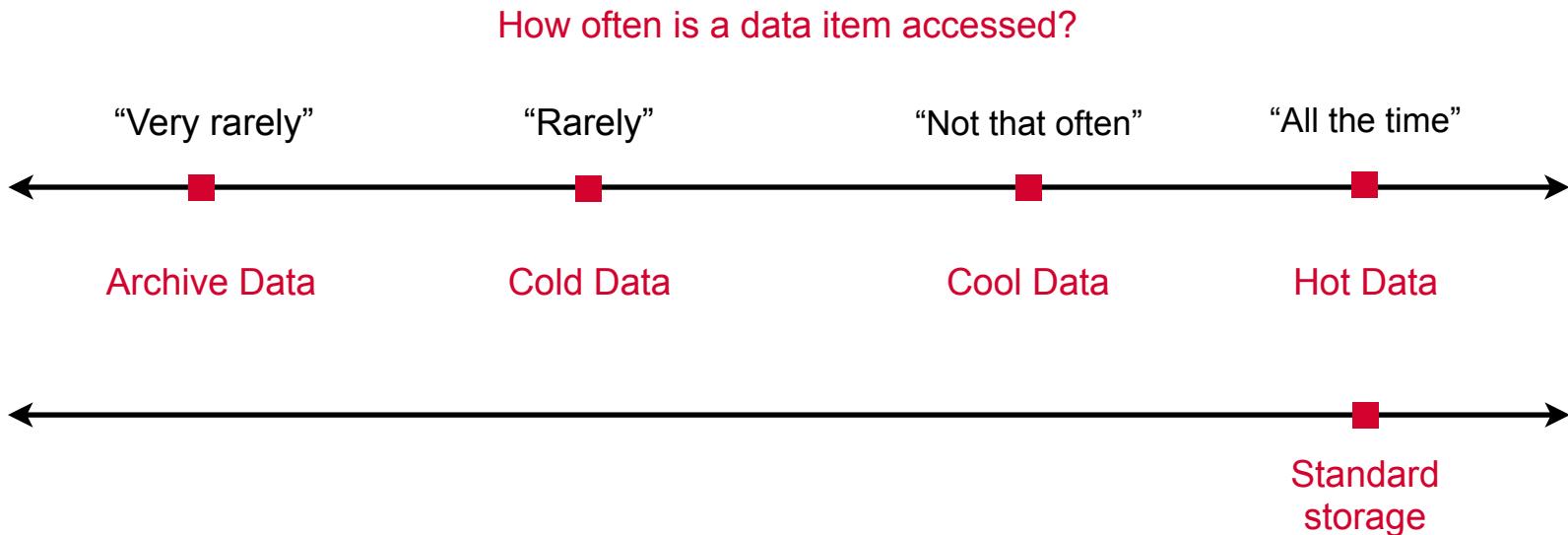


# GCS Storage Classes



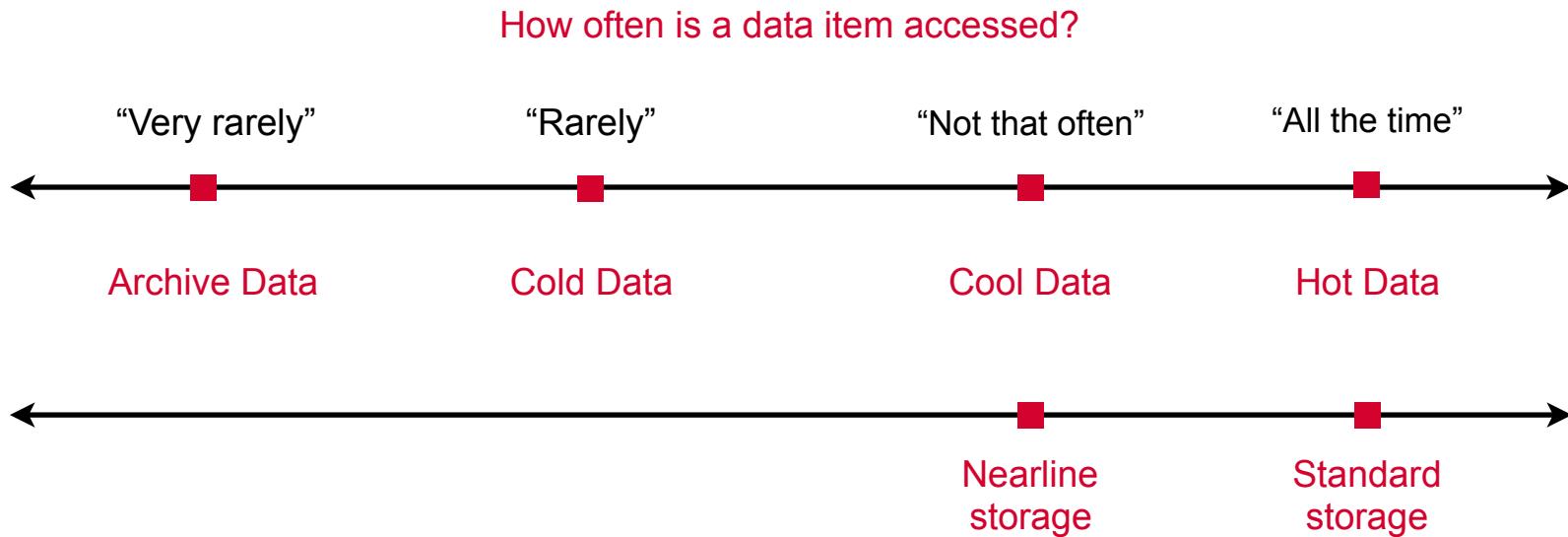


# GCS Storage Classes



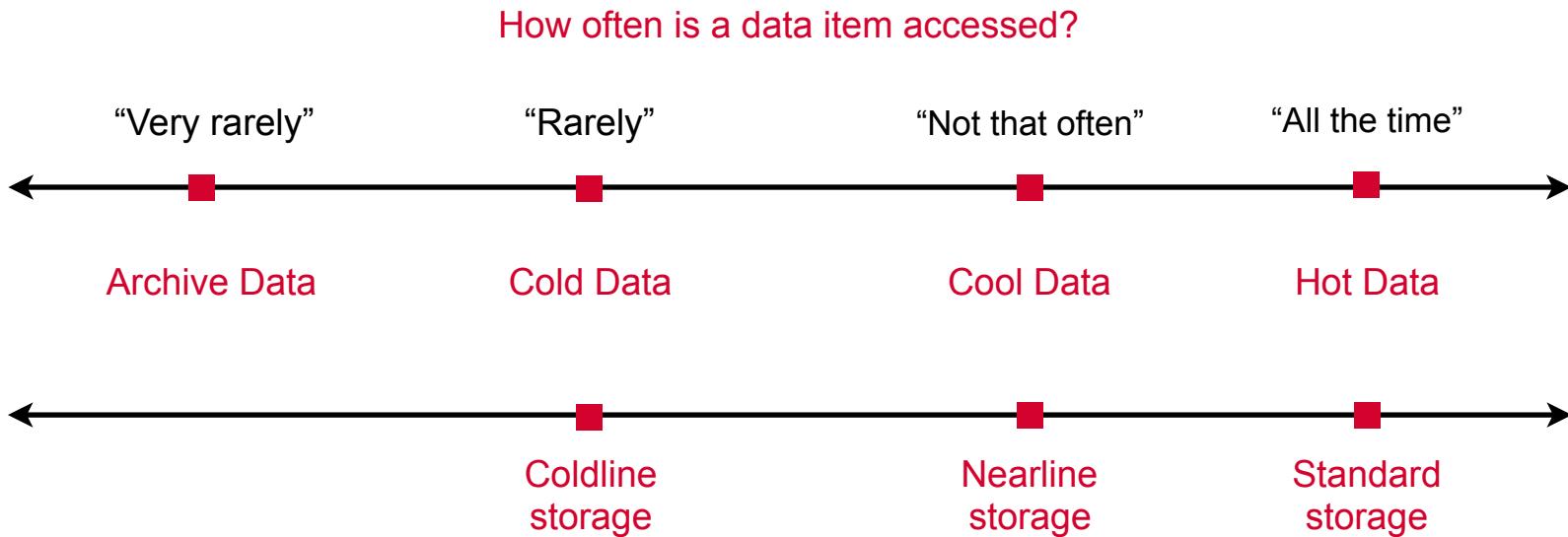


# GCS Storage Classes



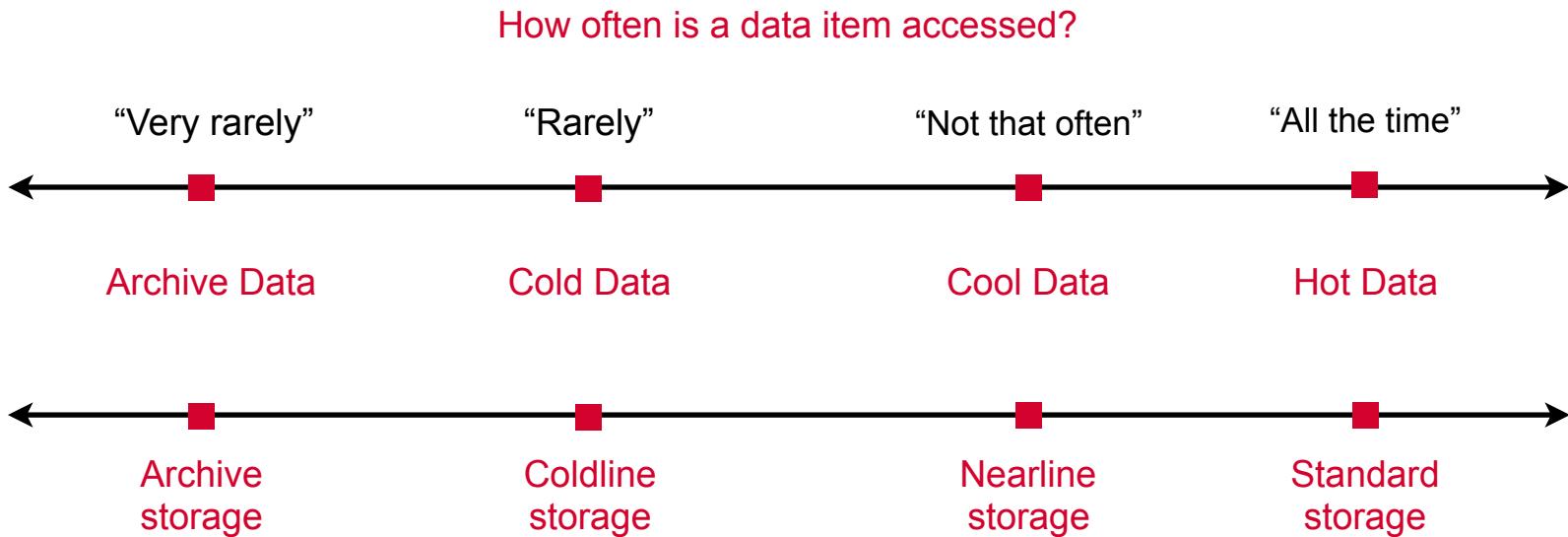


# GCS Storage Classes



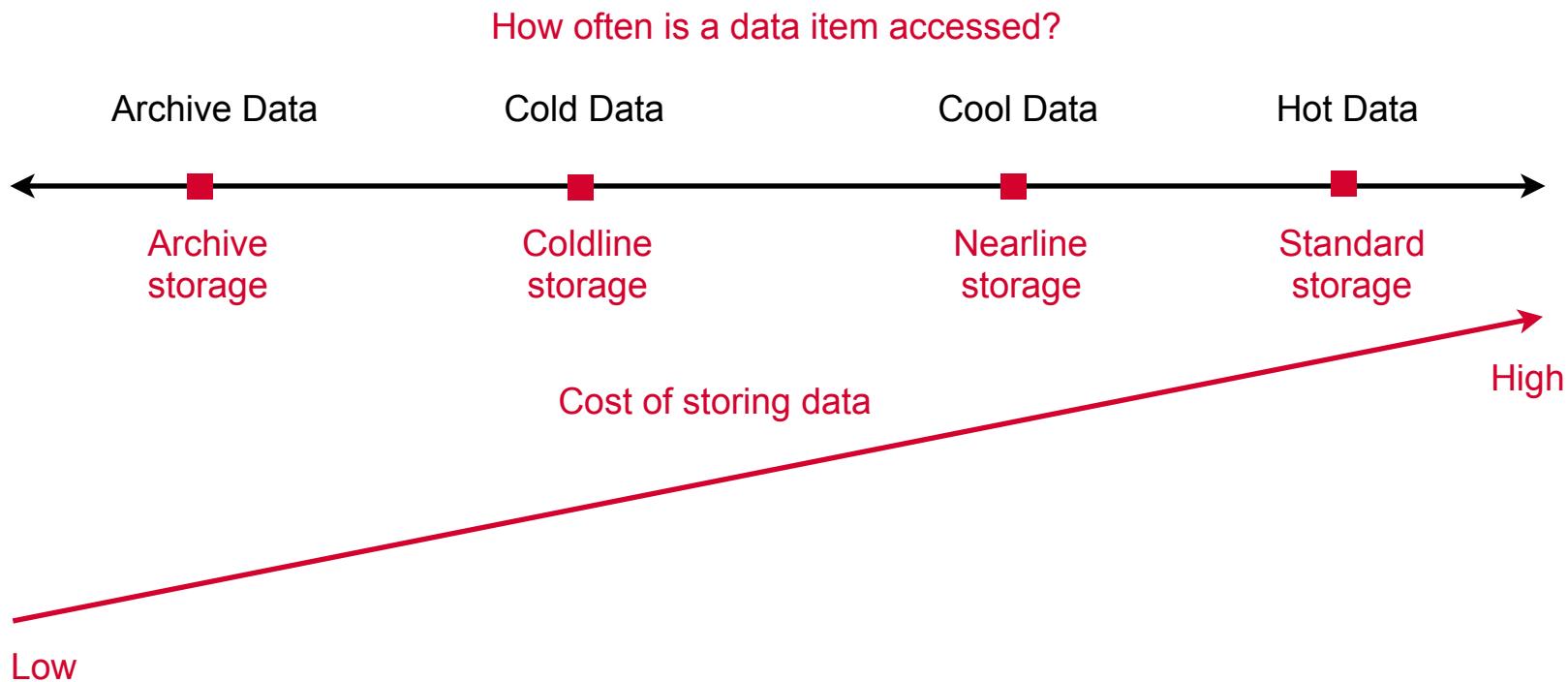


# GCS Storage Classes



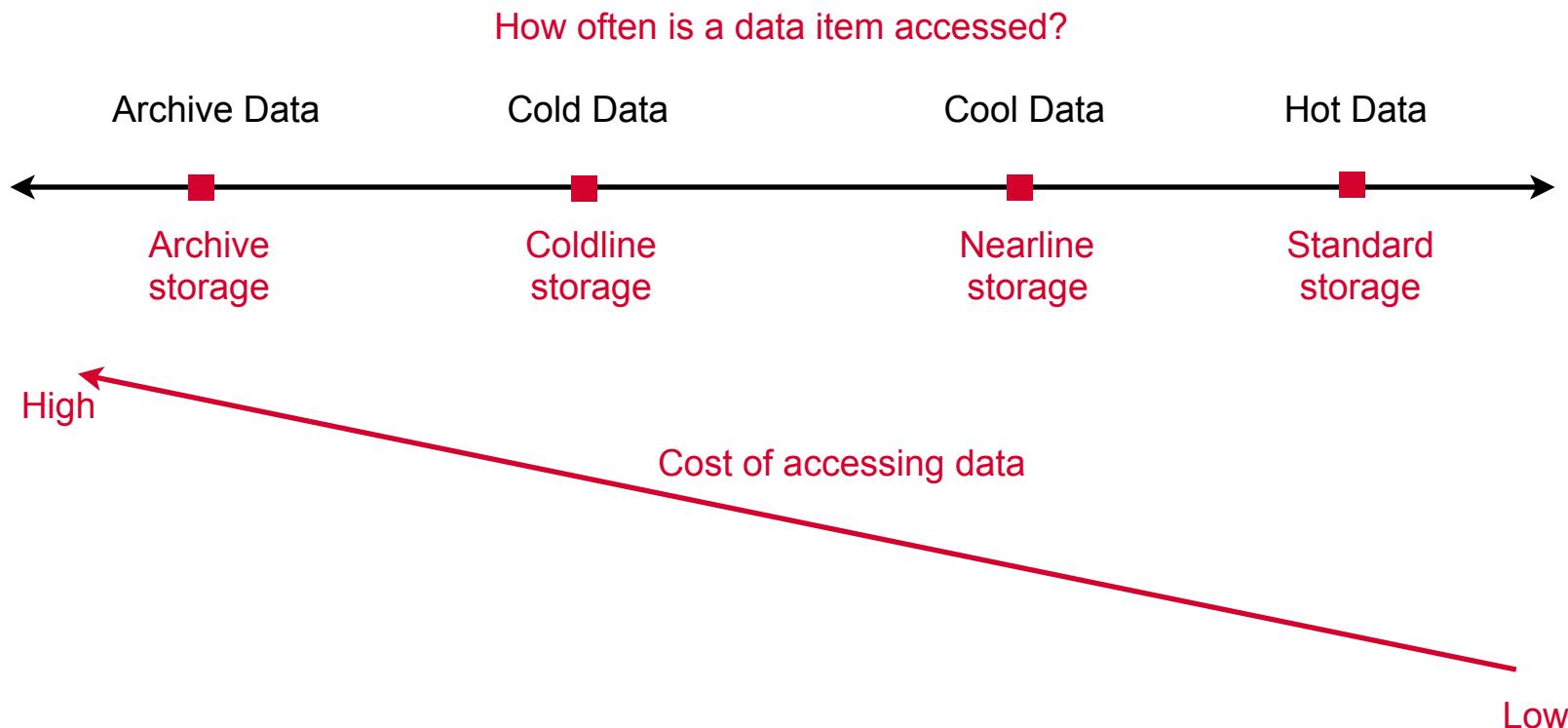


# GCS Storage Classes



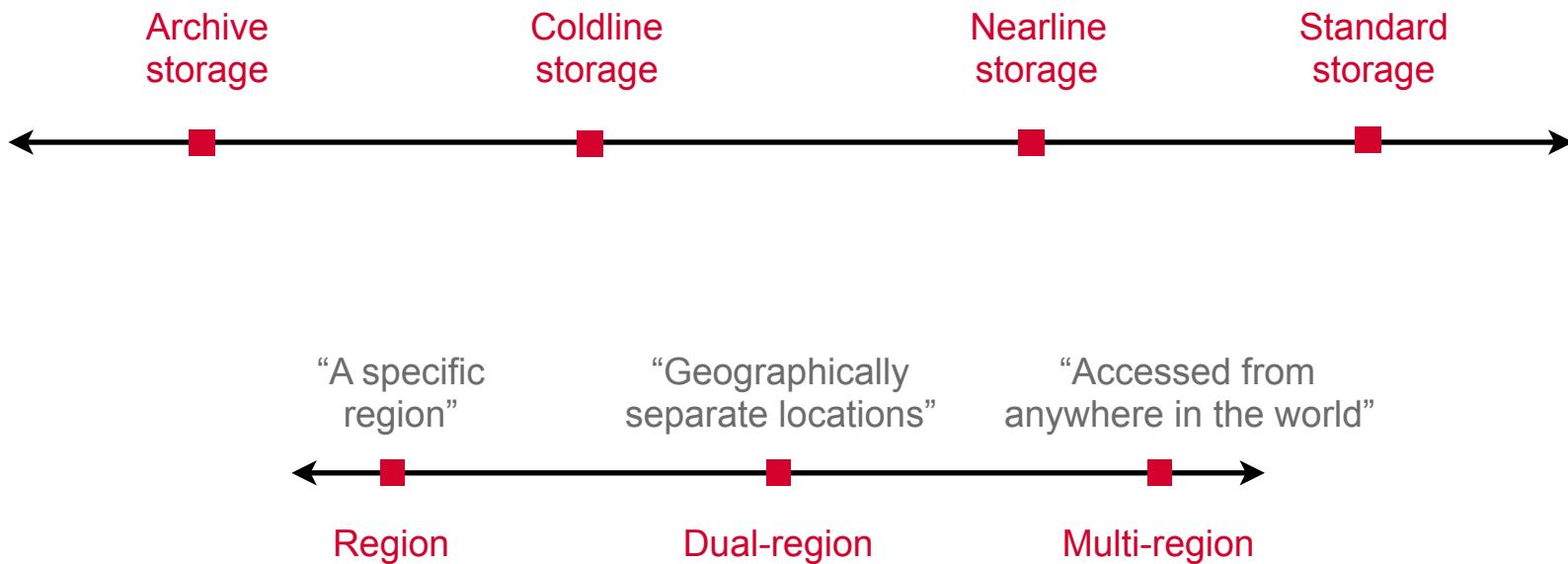


# GCS Storage Classes





# All Storage Classes





# Autoclass

**Moves data that is not accessed to colder storage classes to reduce cost**

**Moves data that is accessed to standard storage to optimize cost of future access**



Availability

Storage Costs

Retrieval Costs

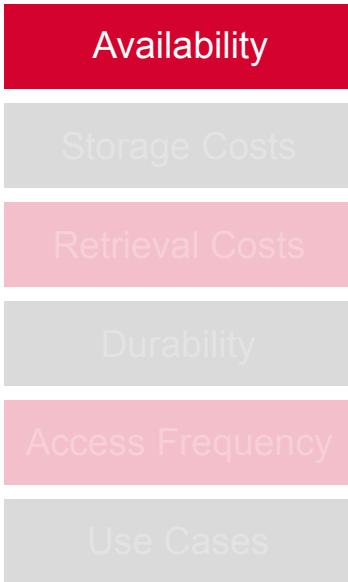
Durability

Access Frequency

Use Cases

Different storage classes represent different trade-offs

Several parameters along which to compare



Storage Class	Availability
Standard storage (dual and multi-regional)	99.95%
Standard storage (regional)	99.9%
Nearline (regional)	99.0%
Coldline (regional)	99.0%

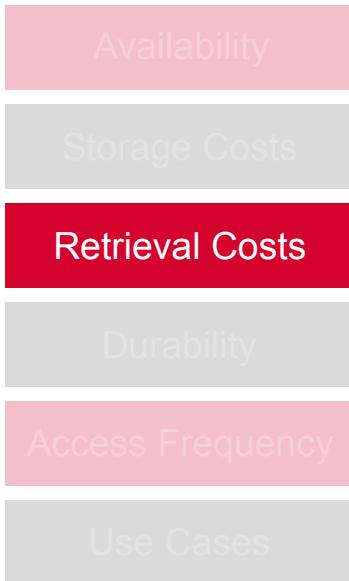


**Dual-region and multi-region buckets are tied  
to multi-regional locations: US, EU and Asia**

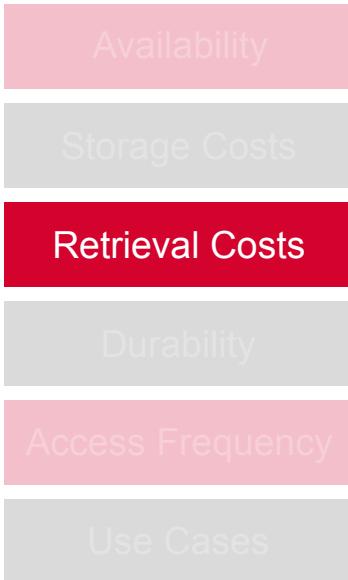
**Helps adhere to data storage regulations in  
the US and EU**



Storage Class	Storage Cost (cents/GB/month)
Standard	2.6
Nearline	1.0
Coldline	0.7
Archive	0.24



Storage Class	Retrieval Cost (cents/GB)
Standard	<b>None</b>
Nearline	1.0
Coldline	2.0
Archive	5.0



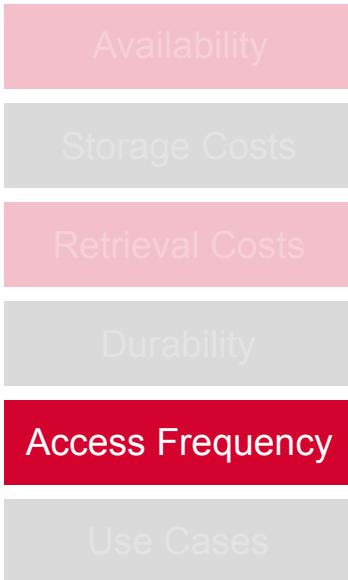
Storage Class	Minimum Commitment
Standard	<b>None</b>
Nearline	30 days*
Coldline	90 days*
Archive	365 days*

\*Early deletion will incur charges

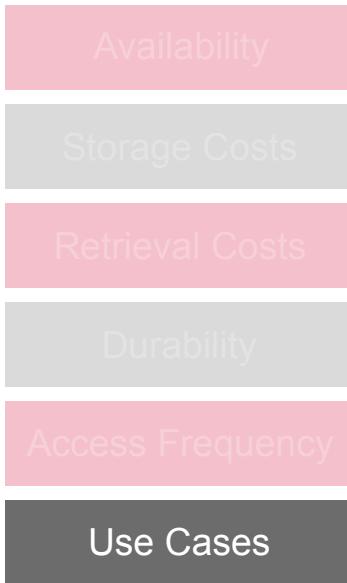


Storage Class	Durability
Standard	99.999999999%
Nearline	99.999999999%
Coldline	99.999999999%
Archive	99.999999999%

“11 nines”



Storage Class	Access Frequency
Standard	Daily
Nearline	Monthly
Coldline	Quarterly
Archive	Less than once a year



Storage Class	Access Frequency
Standard storage (dual and multi-regional)	Serving websites, interactive workloads, mobile and gaming applications
Standard storage (regional)	Access from Compute Engine VMs or Dataproc cluster
Nearline	Data backup, disaster recovery, archival storage
Coldline/Archive	Legal or regulatory needs; <b>also disaster recovery where recovery time is important</b>



# Autoclass

**Moves data that is not accessed to colder storage classes to reduce cost**

**Moves data that is accessed to standard storage to optimize cost of future access**



# Object Versioning

- Needs to be enabled for bucket
- Once enabled, bucket creates archived versions of each object
- Whenever live object is overwritten or deleted
- Version with unique **generation number** is created
- Each copy charged separately





# Object Lifecycle Management

- Can automatically specify changes to object storage class
  - “Change from regional to nearline after 30 days”
  - “Delete all data created before 1/8/2018”
  - “Delete all but 2 most recent versions”





# Encryption

- Encrypted even at rest
- Default: Google generates keys
- Can use CSEK
  - Customer Supplied Encryption Key





# Transfer Appliance

- **Physical, high-capacity storage device** that enables large-scale data migration from your on-premises environment to Google Cloud
- Securely transfer massive amounts of data without relying on network connectivity.
- Data is **encrypted** at rest using AES-256 encryption before it's written to the appliance.
- Google ships the Transfer Appliance to your location.
- You simply connect it to your network, upload your data using standard protocols (like NFS or SMB), and ship it back to Google.
- **Choose when you have to transfer a large amount of data to Google (100 TB+)**





# Storage Transfer Service

- **Fully managed service** to transfer data to Google Cloud from a variety of sources
  - Amazon S3 buckets
  - Azure Blob Storage
  - On-premise file systems
- **Choose over Transfer appliance for small to medium datasets (< 100TB)**
- Can choose for larger datasets when you have sufficient time to perform the transfer (over days or weeks)



# Cloud Storage

Your application frequently accesses files stored in a Cloud Storage bucket for the first 30 days after they are created. After this period, you no longer need frequent access but would like to move the files to a lower-cost storage class suitable for disaster recovery purposes. How should you configure this to minimize costs?

- A. Set up a lifecycle rule to automatically delete files older than 30 days.
- B. Configure a lifecycle policy to move files older than 30 days to Archive Storage for disaster recovery.
- C. Manually move files older than 30 days to Coldline Storage.
- D. Set up a lifecycle policy to transition files older than 30 days to Nearline Storage for disaster recovery.



# Cloud Storage

Your application frequently accesses files stored in a Cloud Storage bucket for the first 30 days after they are created. After this period, you no longer need frequent access but would like to move the files to a lower-cost storage class suitable for disaster recovery purposes. How should you configure this to minimize costs?

- A. Set up a lifecycle rule to automatically delete files older than 30 days.
- B. Configure a lifecycle policy to move files older than 30 days to Archive Storage for disaster recovery.**
- C. Manually move files older than 30 days to Coldline Storage.
- D. Set up a lifecycle policy to transition files older than 30 days to Nearline Storage for disaster recovery.



# Cloud Storage

You are planning to migrate your company's contract files from local servers to Google Cloud Storage. The files have the following storage requirements:

- Contracts must be retained for at least 7 years.
- Up to 3 versions of the same contract must be kept to accommodate updates and revisions.
- Files older than 180 days should be moved to a lower-cost storage class to reduce expenses.

What should you do to meet these requirements?

- A. Set up a lifecycle policy to retain files for 7 years, enable versioning for 3 revisions, and transition files older than 180 days to Nearline Storage.
- B. Use a lifecycle rule to delete files after 180 days, and enable versioning to store 3 versions of each file.
- C. Store files in Coldline Storage from the beginning, and manually manage file versions every 180 days.
- D. Enable versioning for 7 years, keep 3 versions of the contract files, and transition files older than 180 days to Archive Storage.



# Cloud Storage

You are planning to migrate your company's contract files from local servers to Google Cloud Storage. The files have the following storage requirements:

- Contracts must be retained for at least 7 years.
- Up to 3 versions of the same contract must be kept to accommodate updates and revisions.
- Files older than 180 days should be moved to a lower-cost storage class to reduce expenses.

What should you do to meet these requirements?

- A. Set up a lifecycle policy to retain files for 7 years, enable versioning for 3 revisions, and transition files older than 180 days to Nearline Storage.**
- B. Use a lifecycle rule to delete files after 180 days, and enable versioning to store 3 versions of each file.
- C. Store files in Coldline Storage from the beginning, and manually manage file versions every 180 days.
- D. Enable versioning for 7 years, keep 3 versions of the contract files, and transition files older than 180 days to Archive Storage.



# Cloud Storage

You are setting up data storage for a research team that handles large datasets for real-time analysis. The data needs to be accessed frequently throughout the day for critical processing tasks, and the team is located in Berlin, Germany. Your goal is to optimize storage costs while ensuring fast and reliable access for the team.

What should you do?

- A. Configure multi-region storage to improve redundancy, and use Nearline storage for cost savings.
- B. Set up regional storage in the region closest to the users and choose the Standard storage class for frequent access.
- C. Configure Coldline storage in a nearby region to minimize costs while ensuring availability.
- D. Use Archive storage in a multi-region setup to reduce storage costs for frequent access.



# Cloud Storage

You are setting up data storage for a research team that handles large datasets for real-time analysis. The data needs to be accessed frequently throughout the day for critical processing tasks, and the team is located in Berlin, Germany. Your goal is to optimize storage costs while ensuring fast and reliable access for the team.

What should you do?

- A. Configure multi-region storage to improve redundancy, and use Nearline storage for cost savings.
- B. Set up regional storage in the region closest to the users and choose the Standard storage class for frequent access.**
- C. Configure Coldline storage in a nearby region to minimize costs while ensuring availability.
- D. Use Archive storage in a multi-region setup to reduce storage costs for frequent access.

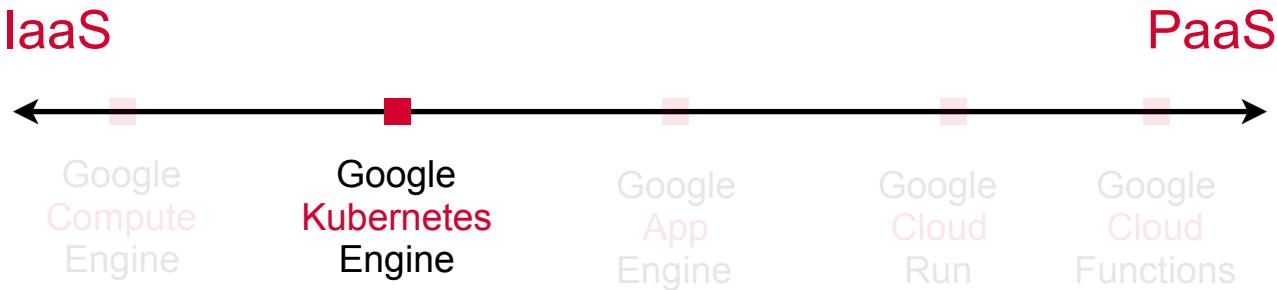


# Containers and Kubernetes





# Google Cloud Compute Choices





# Drawbacks of VMs

- Contain guest OS
  - Introduces platform dependency
  - Bloats image size to GB (apps far smaller)
- Heavyweight
  - Slow to boot up
  - Slow to scale
- Not trivial to migrate
  - VM migration tools needed





# Container

A container image is a lightweight, stand-alone, executable package of a piece of **software that includes everything needed to run it**; code, runtime, system tools, system libraries, settings



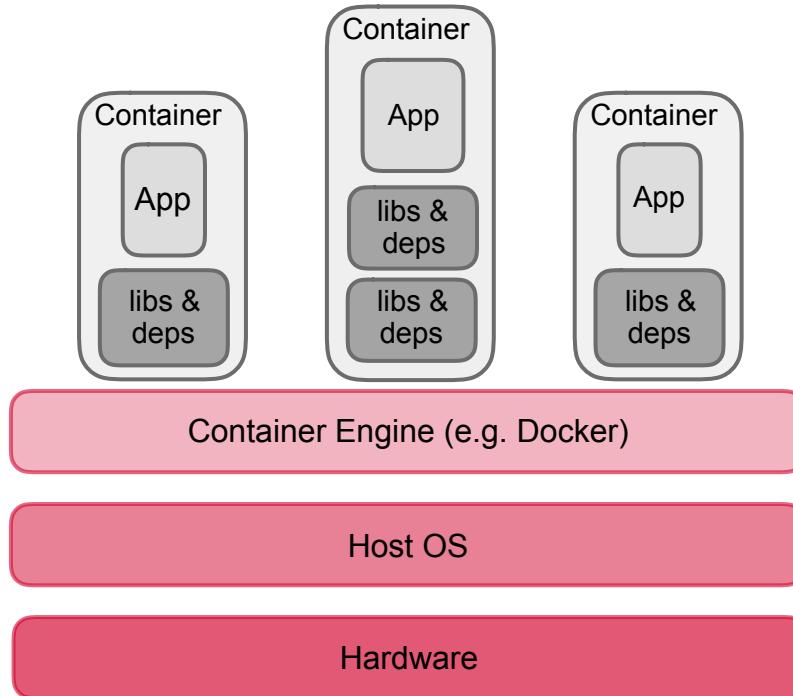
# Container

- Contains applications
- And all of the application's dependencies
- Platform independent
- Runs on layer of abstraction
- Docker Runtime (for Docker containers)





# Modern Workloads on Containers





# Attractions of Containers

- No guest OS
  - Platform independent
  - Considerably smaller than VM images
- Lightweight
  - Small and fast
  - Quick to start
  - Speeds up autoscaling
- Hybrid, multi-cloud
  - Hybrid: Work on-premise and on cloud
  - Multi-cloud: Not tied to any specific cloud platform





# Standalone Container Limitations

- No autohealing
  - Crashed containers won't restart automatically
  - Need higher level orchestration
- No scaling or autoscaling
  - Overloaded containers don't spawn more automatically
  - Need higher level orchestration
- No load balancing
  - Containers can't share load automatically
  - Need higher level orchestration
- No isolation
  - Crashing containers can take each other down
  - Need sandbox to separate them





# Kubernetes

Orchestration technology for containers - convert isolated containers running on different hardware into a cluster



**Kubernetes is the middle-ground  
between IaaS and PaaS in a hybrid,  
multi-cloud world**



# IaaS vs. PaaS

## Infrastructure-as-a-Service

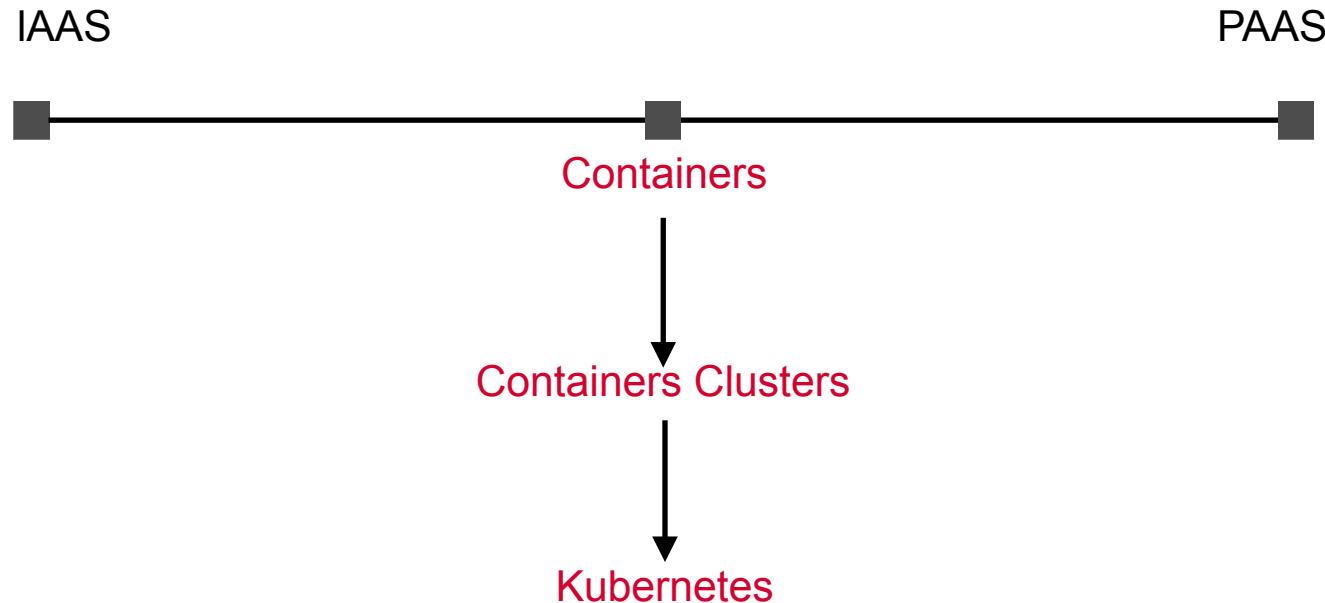
- Heavy operational burden
- Migration is hard

## Platform-as-a-Service

- Provider lock-in
- Migration is very hard



# Compute Choices





# Kubernetes as Orchestrator

- Fault-tolerance
- Autohealing
- Isolation
- Scaling
- Autoscaling
- Load balancing





# Google Kubernetes Engine (GKE)

- Service for working with Kubernetes clusters on GCP
- Runs Kubernetes on GCE VM instances
- Many more abstractions and a lot more support than using plain Kubernetes on-premises

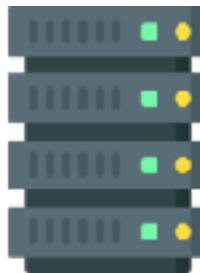
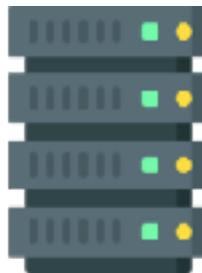
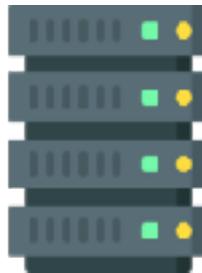




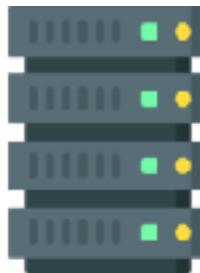
# Kubernetes Clusters



Master node



Worker  
nodes

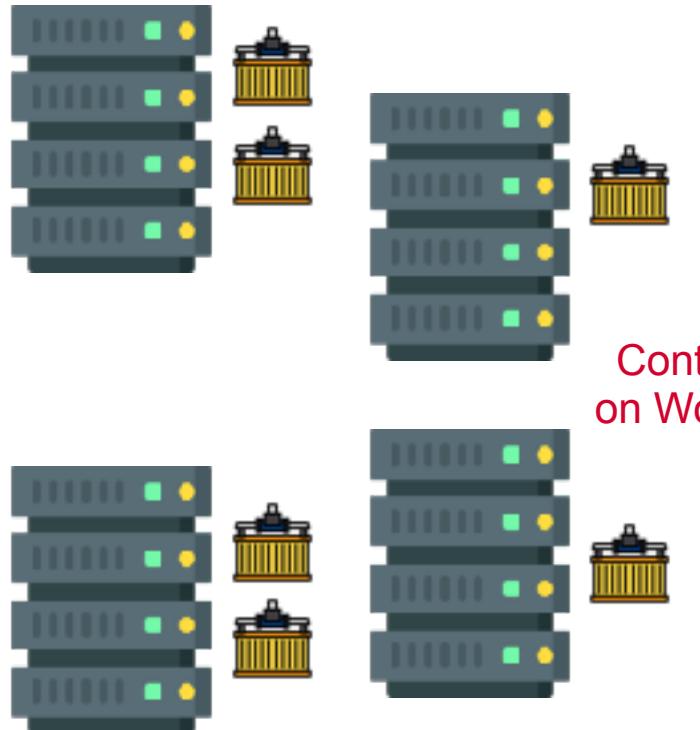




# Kubernetes Clusters



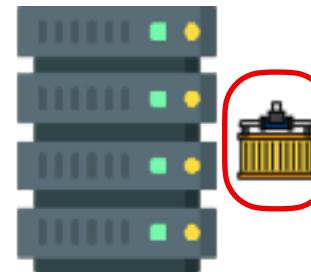
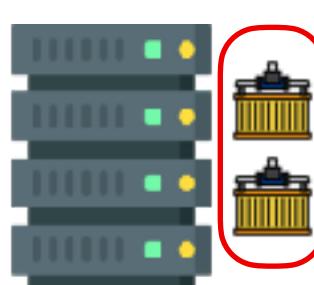
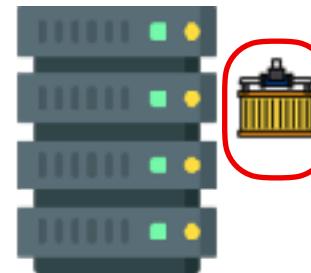
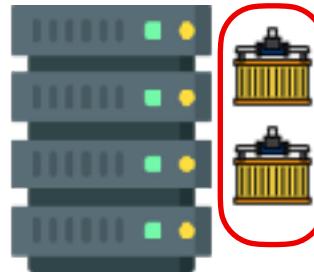
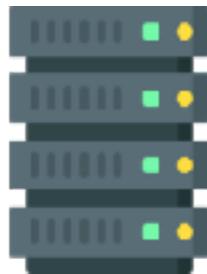
Users interact with  
master node



Containers run  
on Worker nodes



# Containers Deployed in Pods





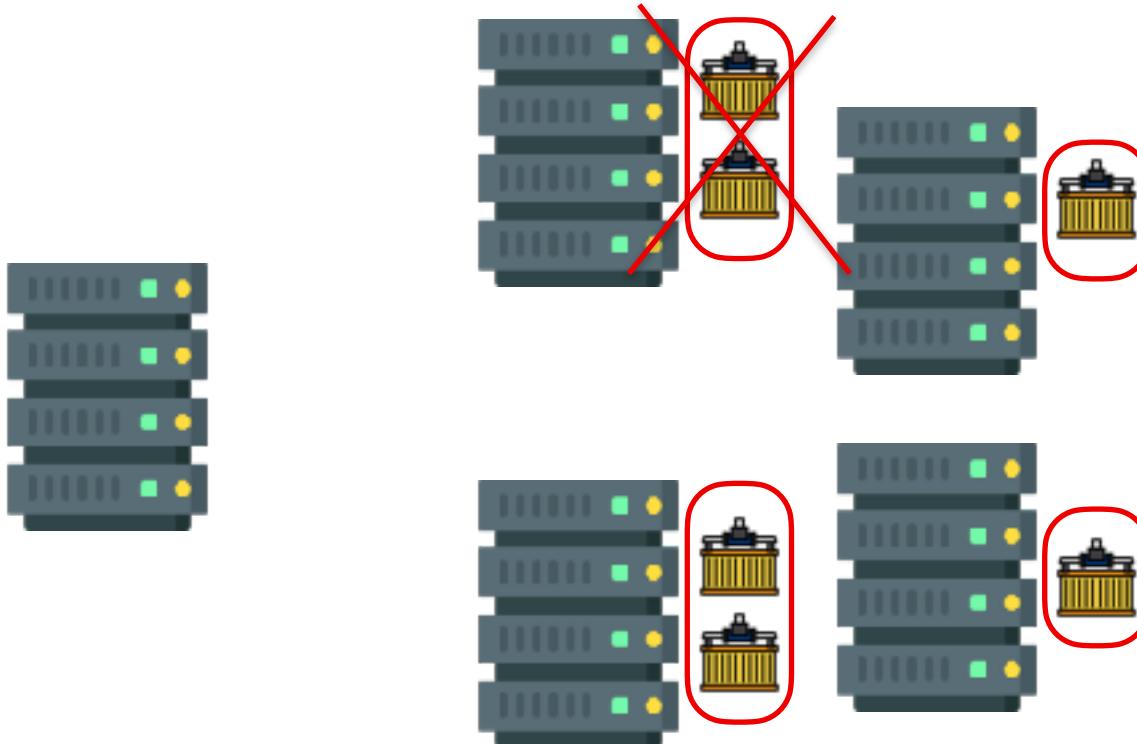
# Kubernetes Pods

- A pod is the **smallest and simplest unit of deployment** in Kubernetes
- Can run a single container or multiple containers
  - Multiple containers in a pod work closely together
  - Communicate through shared memory, use shared resources





# Pods are Ephemeral



Kubernetes replaces pods if they fail, and they may be recreated or moved across nodes. Each pod gets a unique IP, so new pod instances don't retain the same identity as previous ones.



# Nodes



On-premises or cloud VMs on which the containers are run



# Nodes



Run services to run containers e.g. Docker containers,  
communicate with the master node



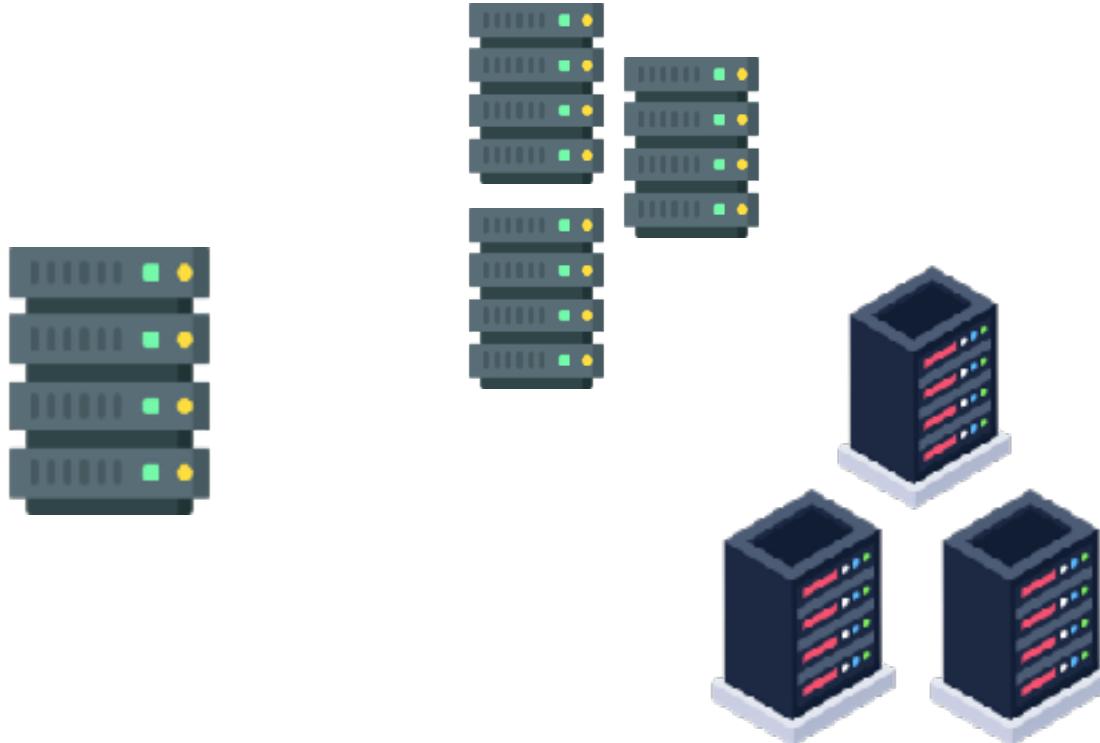
# Node Images



Special operating system optimized to run containers on the  
Google Cloud



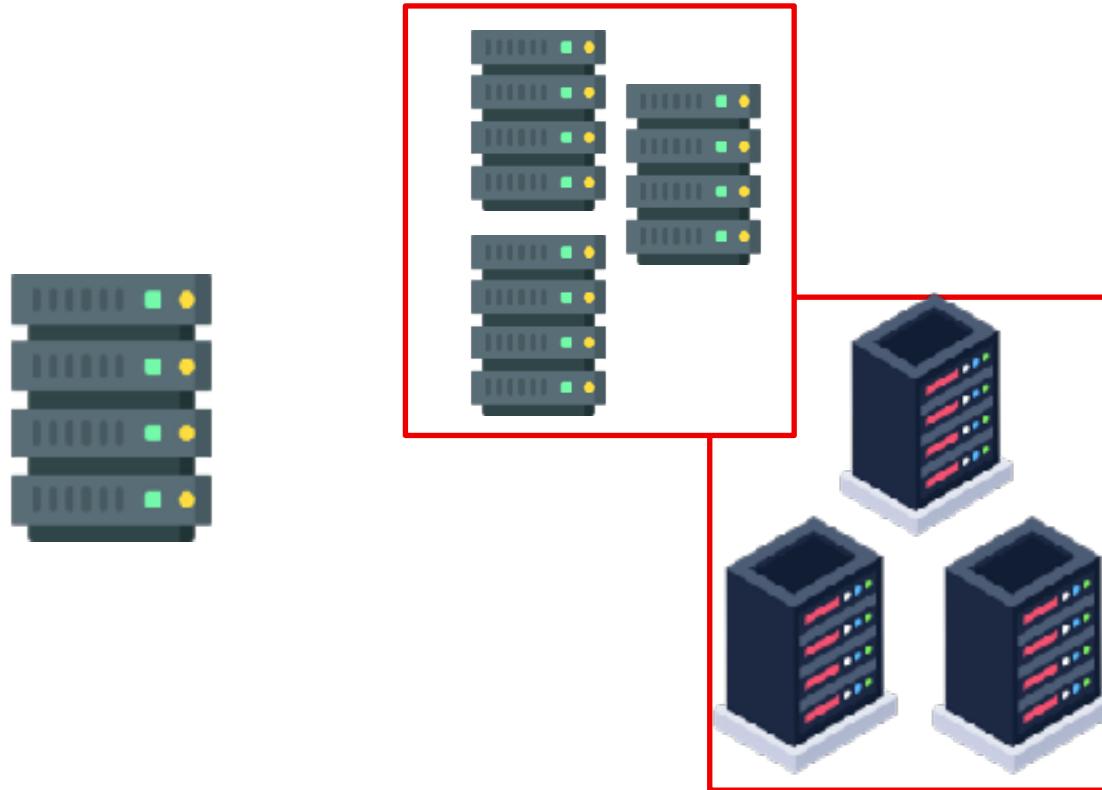
# Node Pools



Groups of nodes in your cluster that have the same configuration settings such as machine type, disk size, and labels



# GKE Clusters Can Have Multiple Node Pools



Each node pool can be optimized for different workloads or operational needs



# Each Node Pool Can be Configured to Autoscale



Optimize resource allocation in a workload specific manner



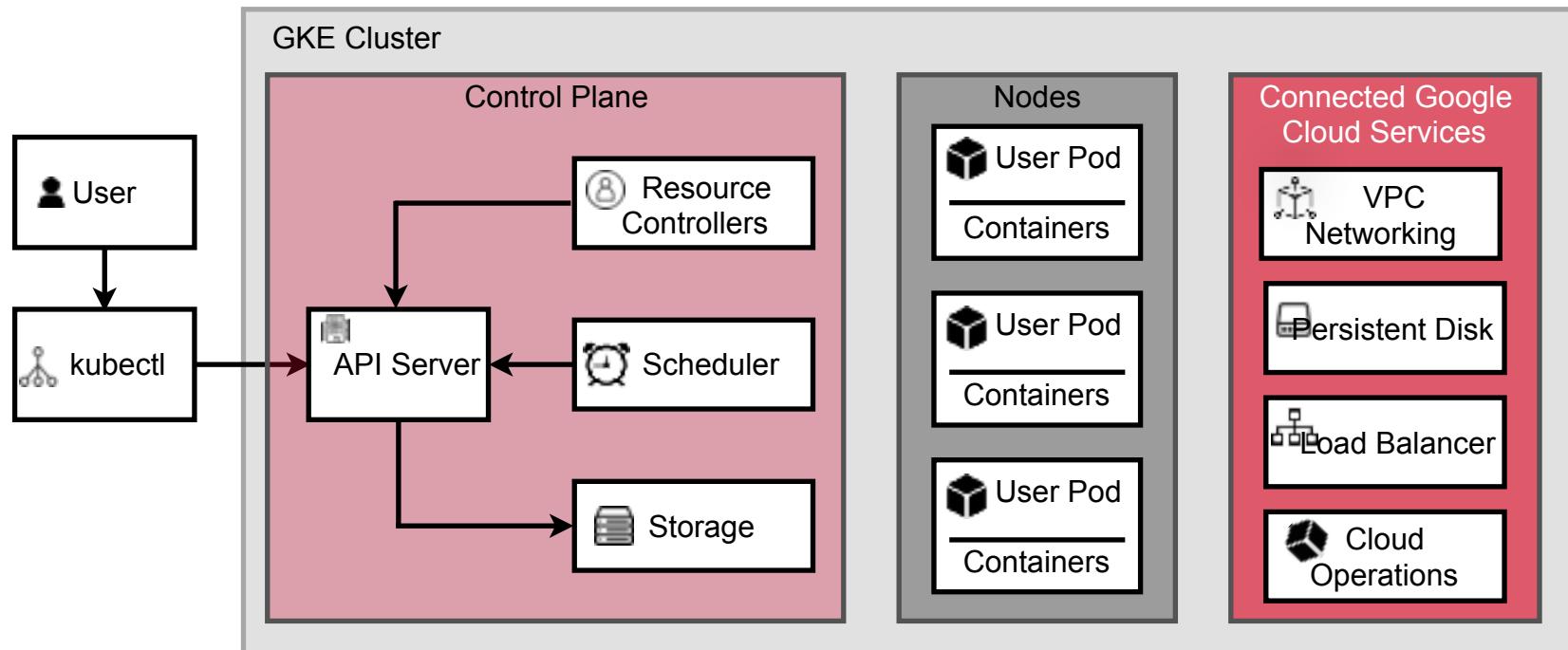
# Uses of Node Pools

- Each node pool has its own **custom configuration**
  - e.g. GPUs, Spot VMs
  - Tailor different node pools for different workloads
- **Workload segregation** using node pools
  - Separate workloads that need high computing power, fault tolerant workloads, critical workloads





# GKE Cluster

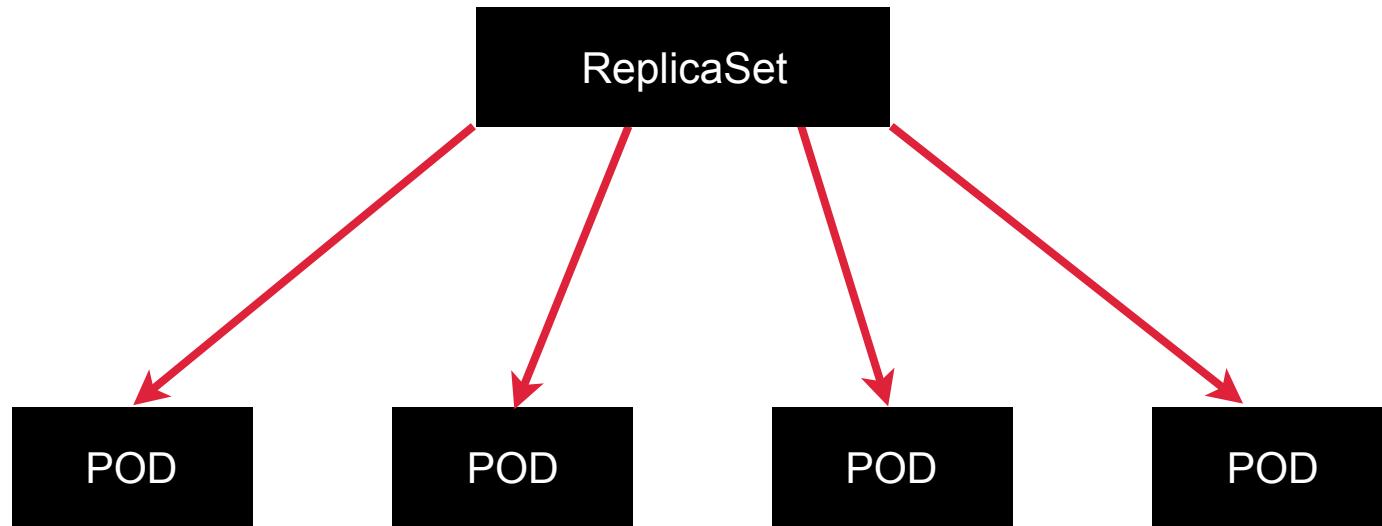




**Kubernetes uses higher level abstractions to  
deal with containers running in the cluster**



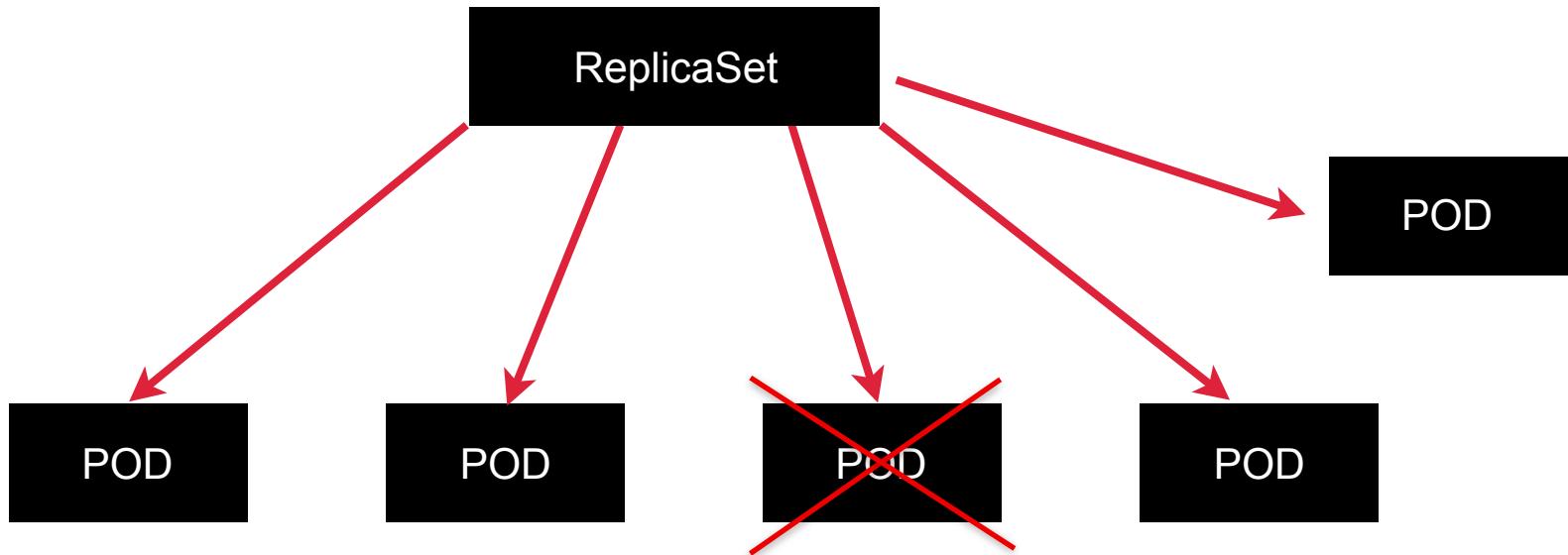
# ReplicaSets



Kubernetes abstraction responsible for maintaining a specified number of identical pod instances running at all times



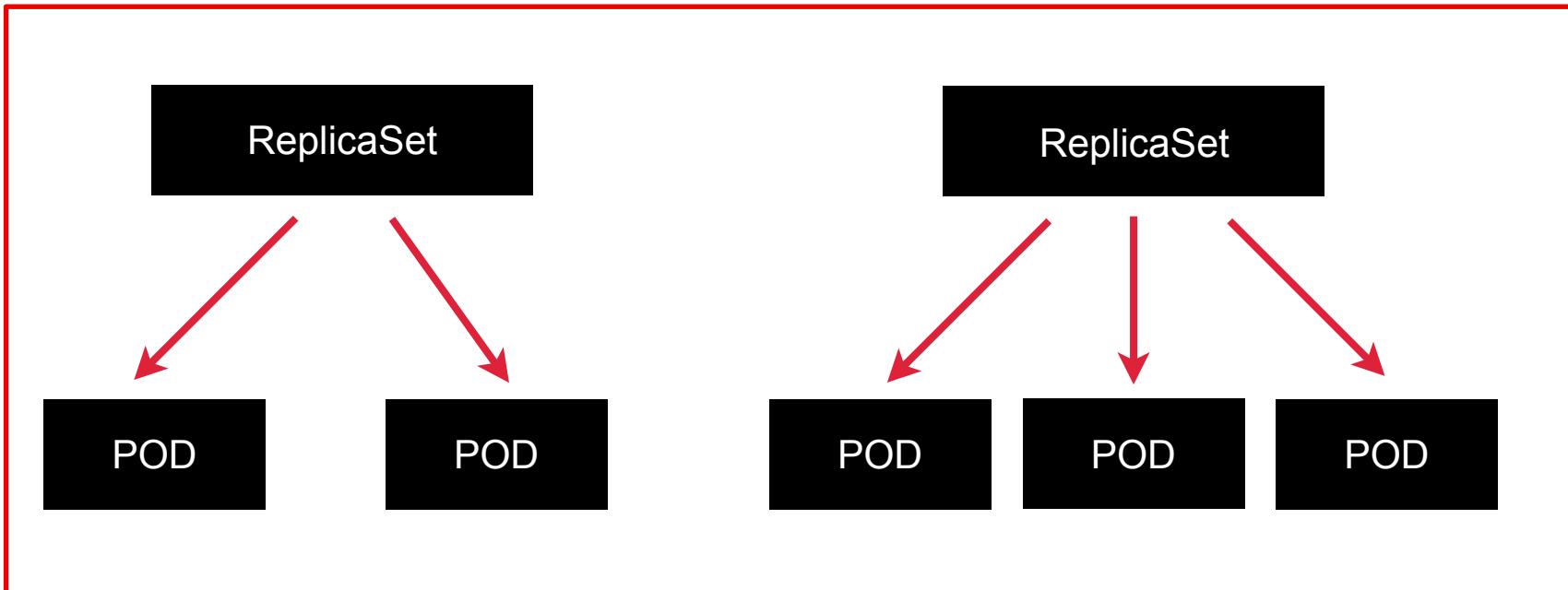
# ReplicaSets



If a pod crashes the ReplicaSet will create a new pod to replace the crashed pod - this maintains the high availability and resilience of the application



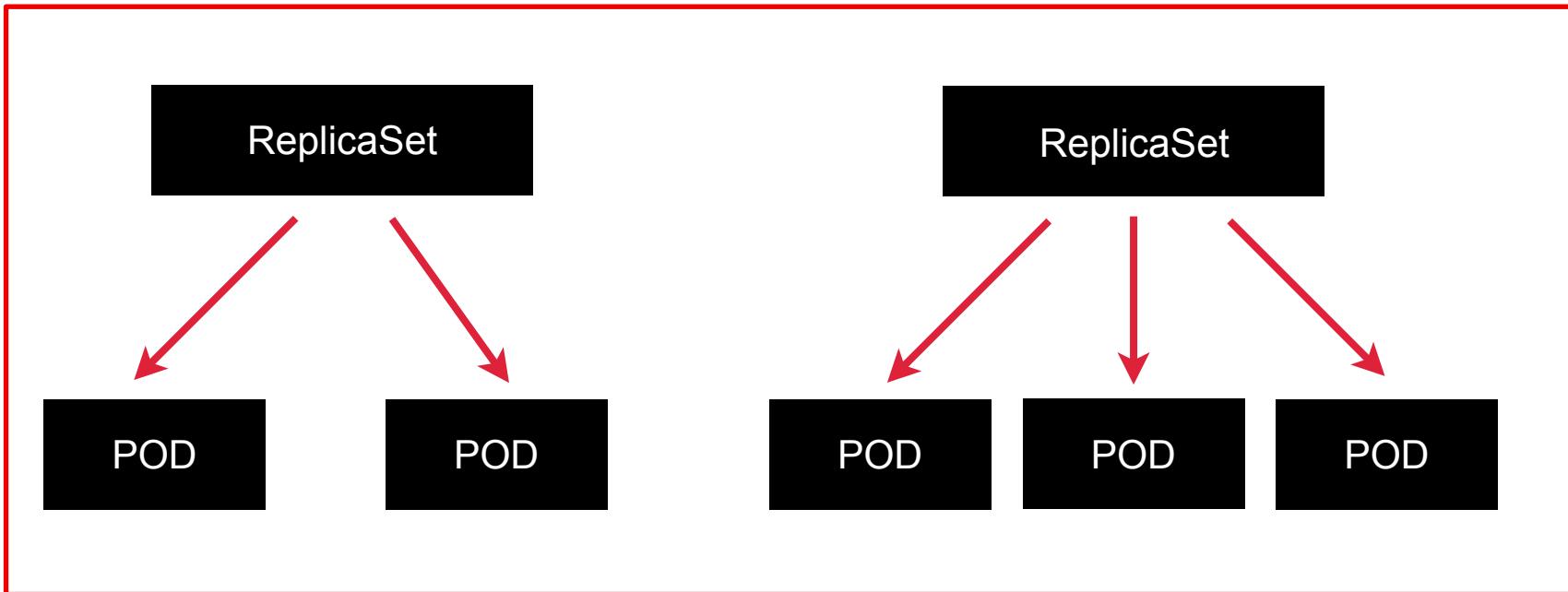
# Deployment



Higher level abstraction that manages ReplicaSets



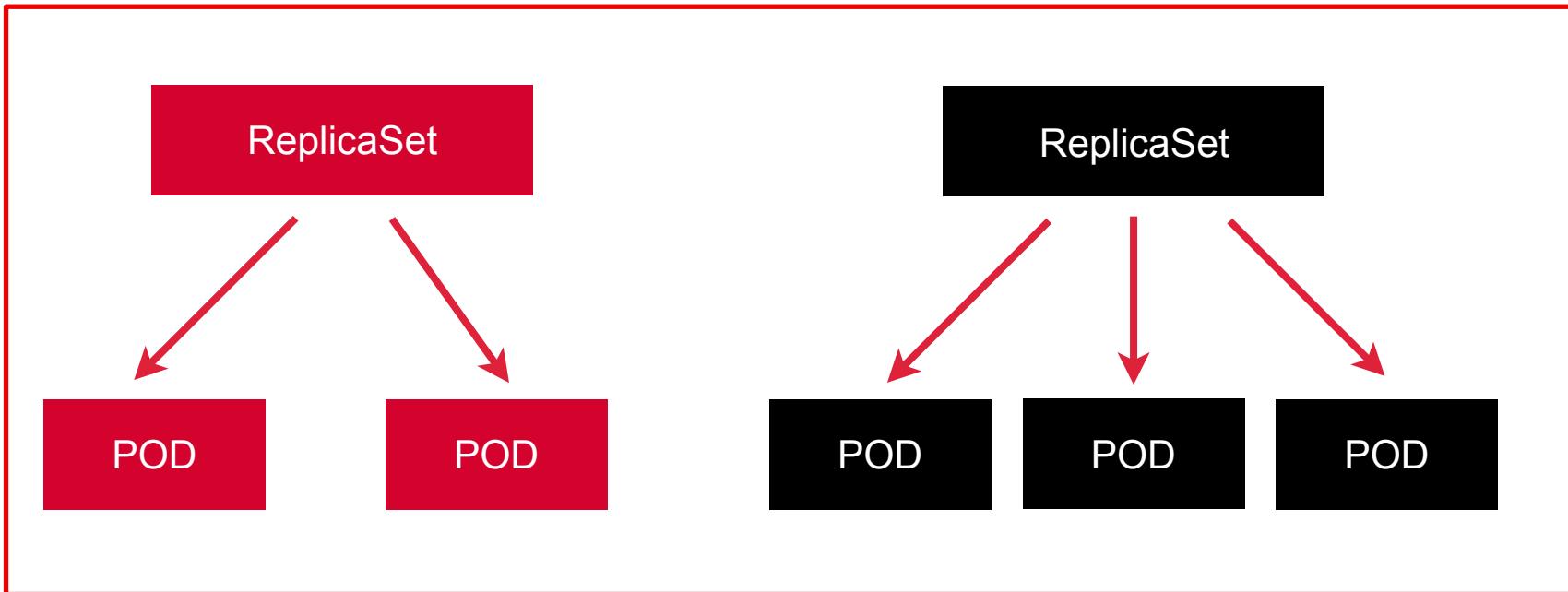
# Deployment – Versioning and Rollback



Adds versioning and rollback functionality to applications



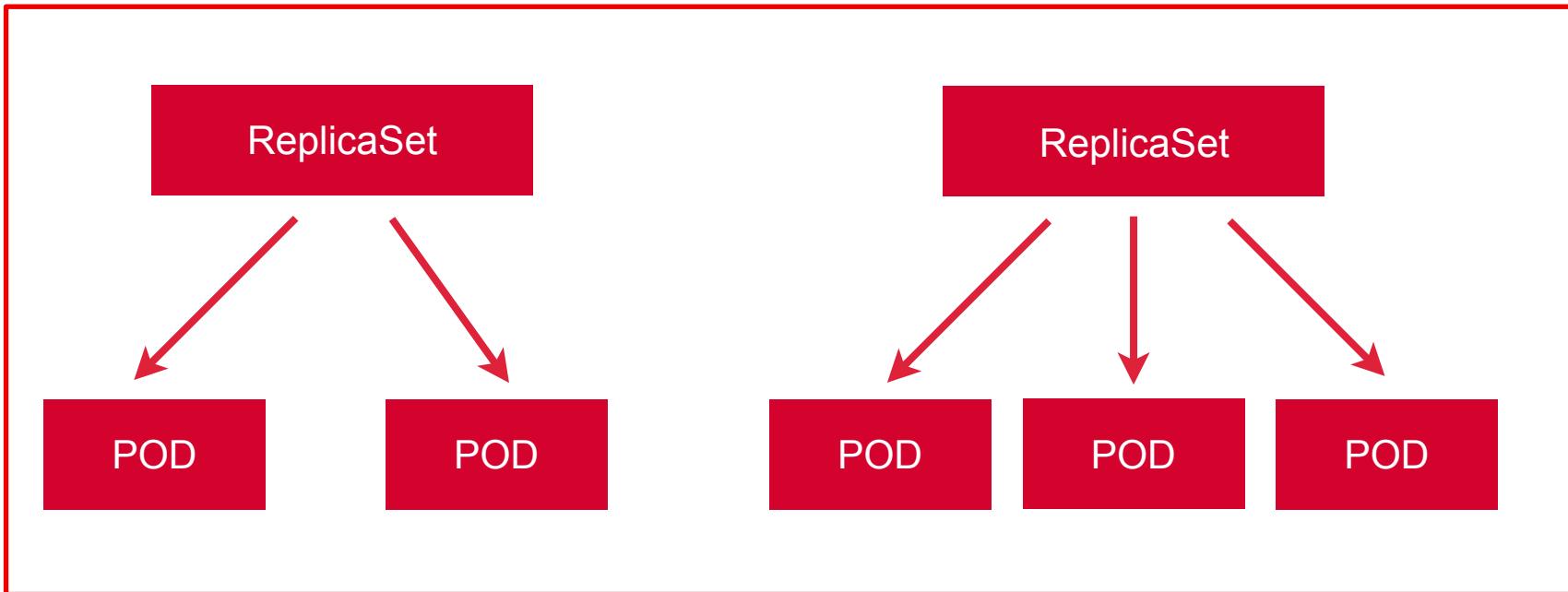
# Deployment – Rolling Updates



When deploying a new version of an application, Kubernetes performs a rolling update, gradually replacing old pods with new ones. This ensures zero downtime by keeping the application available during the update process.



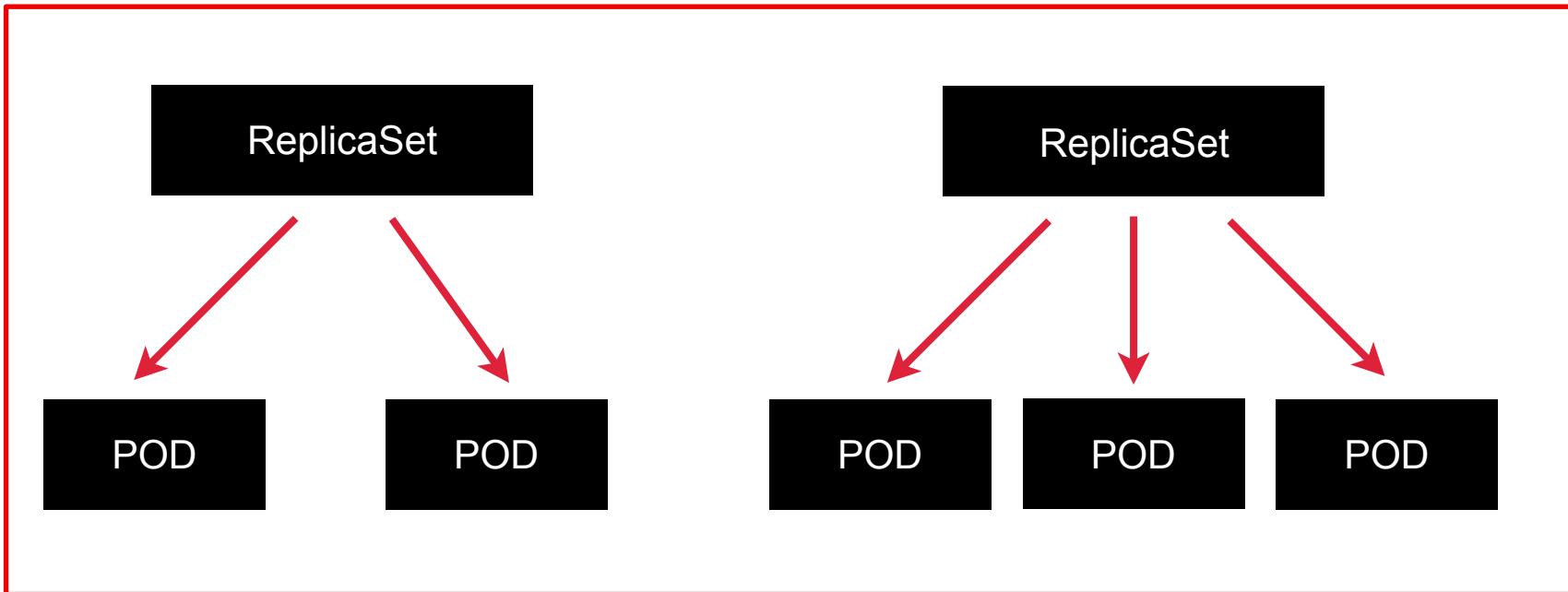
# Deployment – Rolling Updates



If updates cause problems Kubernetes can quickly rollback the Deployment to a prev



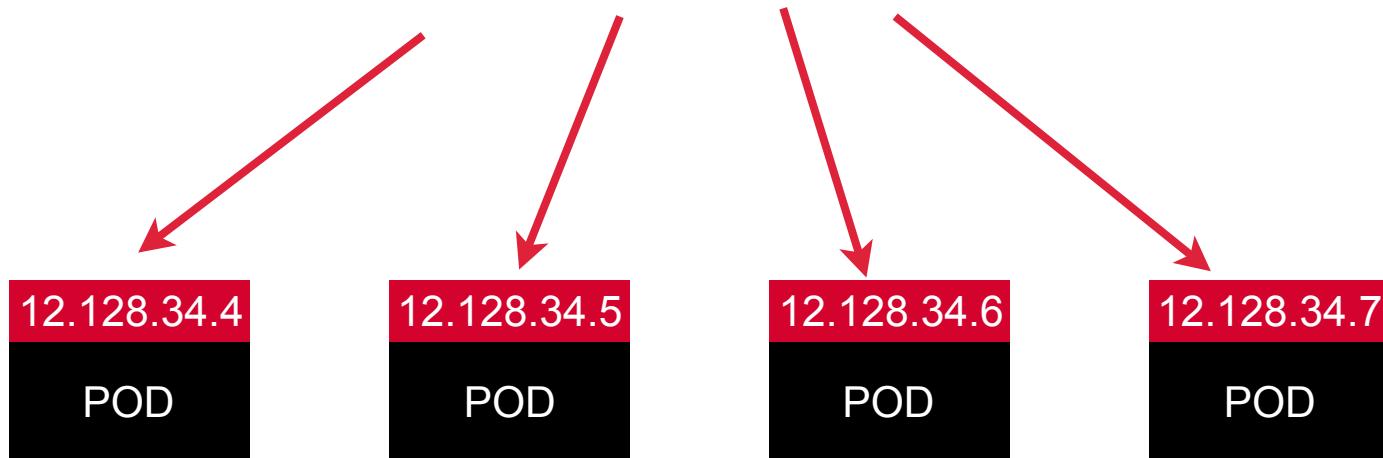
# Deployment – Rollbacks



If updates cause problems Kubernetes can quickly rollback the Deployment to a previous stable version



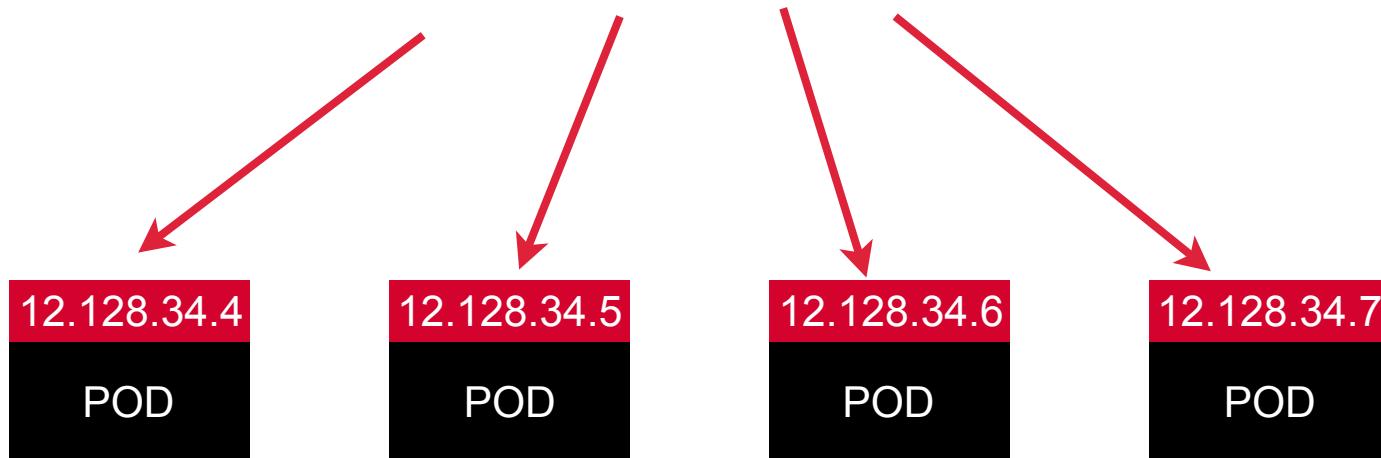
# Pod IP Addresses are Ephemeral



Each time a pod is recreated it will be assigned a new IP address



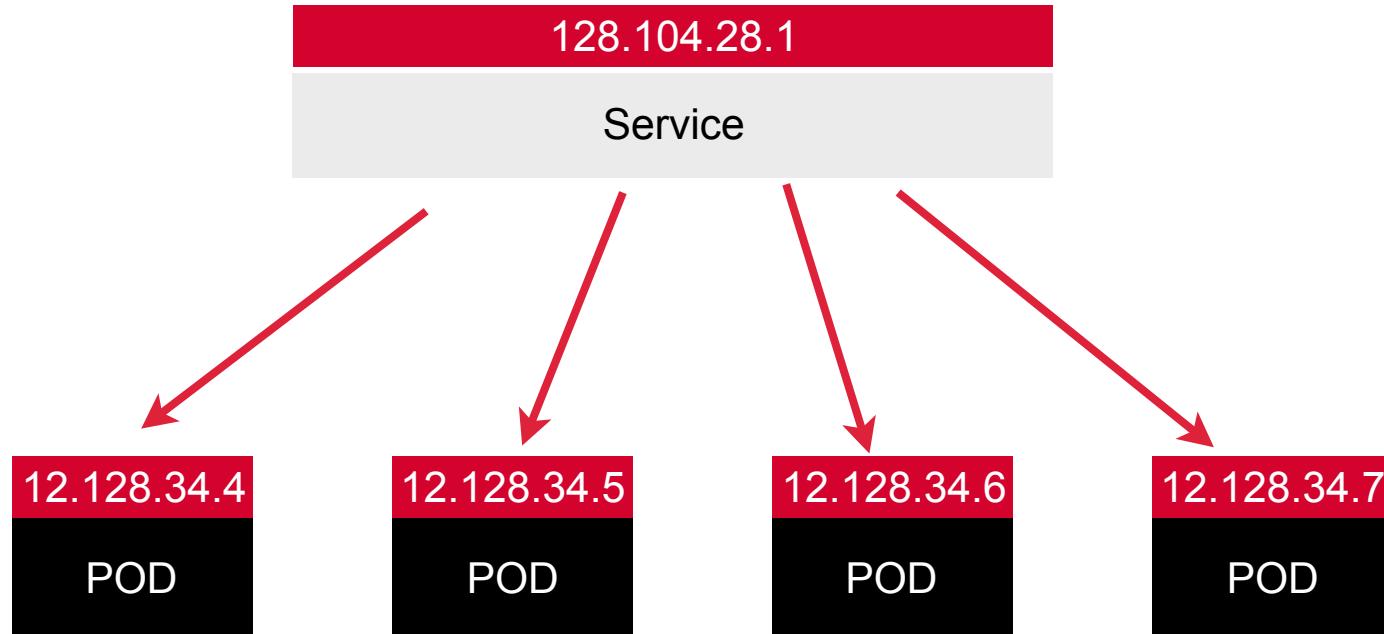
# Pod IP Addresses are Ephemeral



Cannot route traffic to pods using their IP addresses



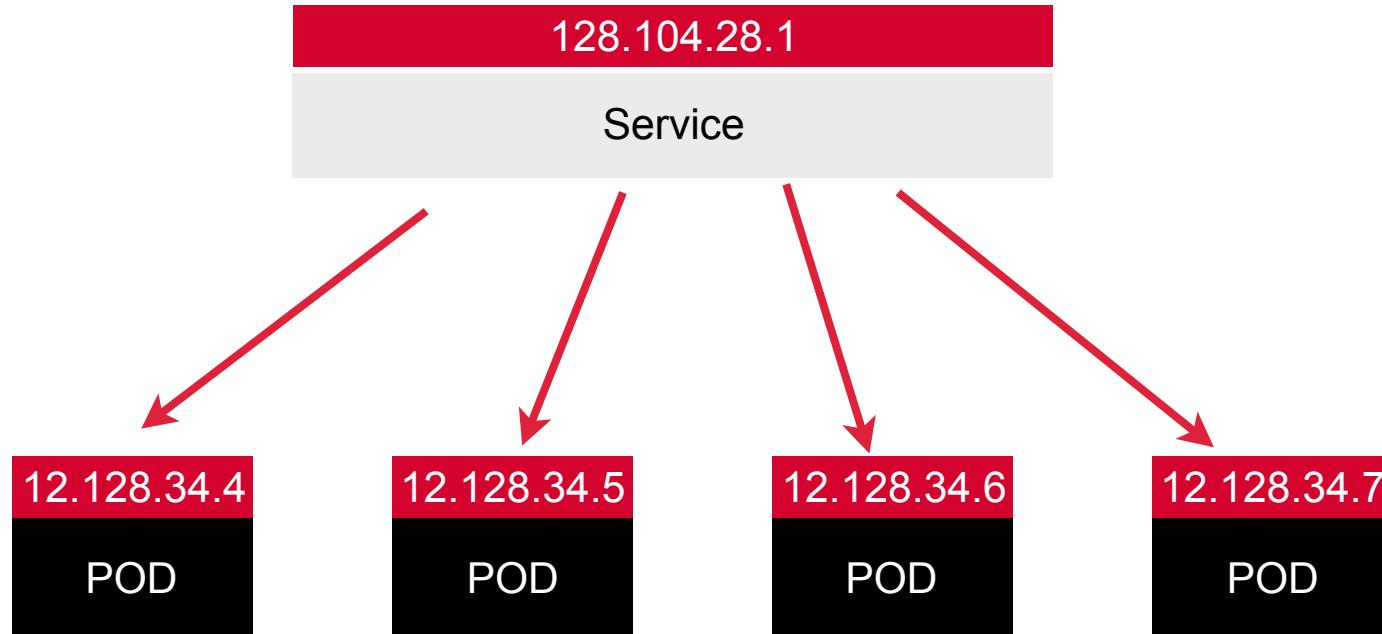
# Service



Provides a consistent IP address or DNS name that remains constant even as the underlying pods are created or destroyed



# Service



Makes it easy for users, other services or pods to connect to the application running in pods



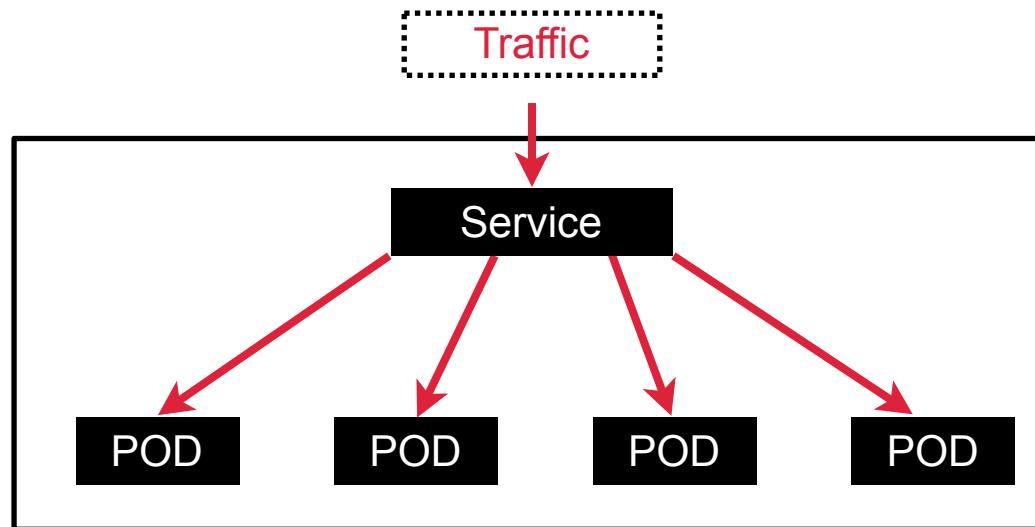
# Types of Services

- **Cluster IP**
  - Exposes the service only inside the cluster accessible to other pods in the cluster
- **NodePort**
  - Exposes the service on a specific port of each node
- **LoadBalancer**
  - Integrates with cloud providers to provision a load balancer making the services accessible outside the cluster





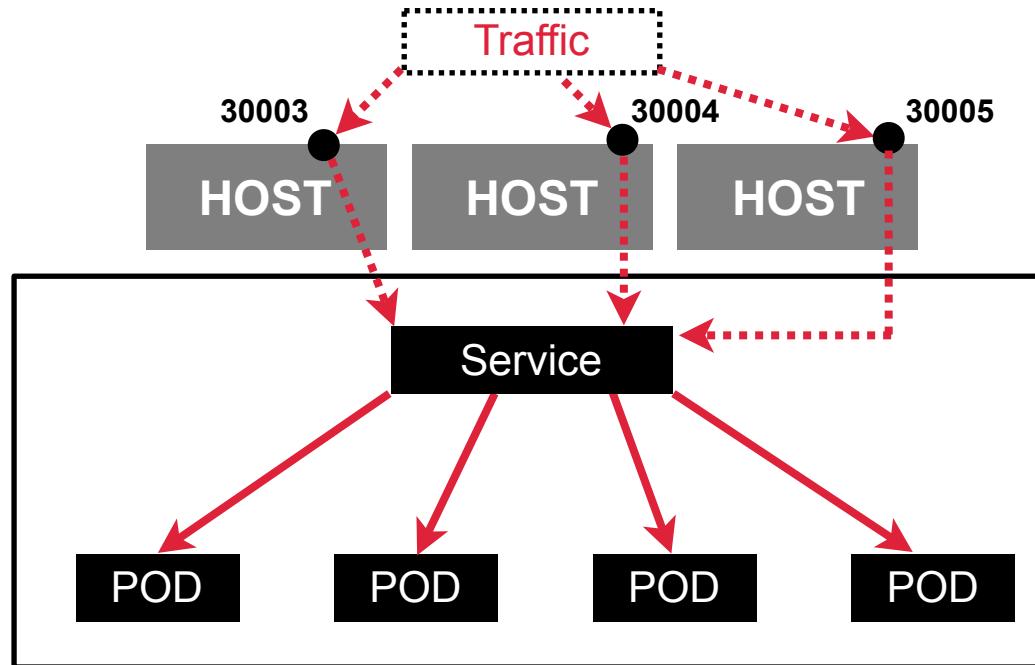
# ClusterIP



Great for internal connectivity between different services in an application that don't need to be exposed to the outside world



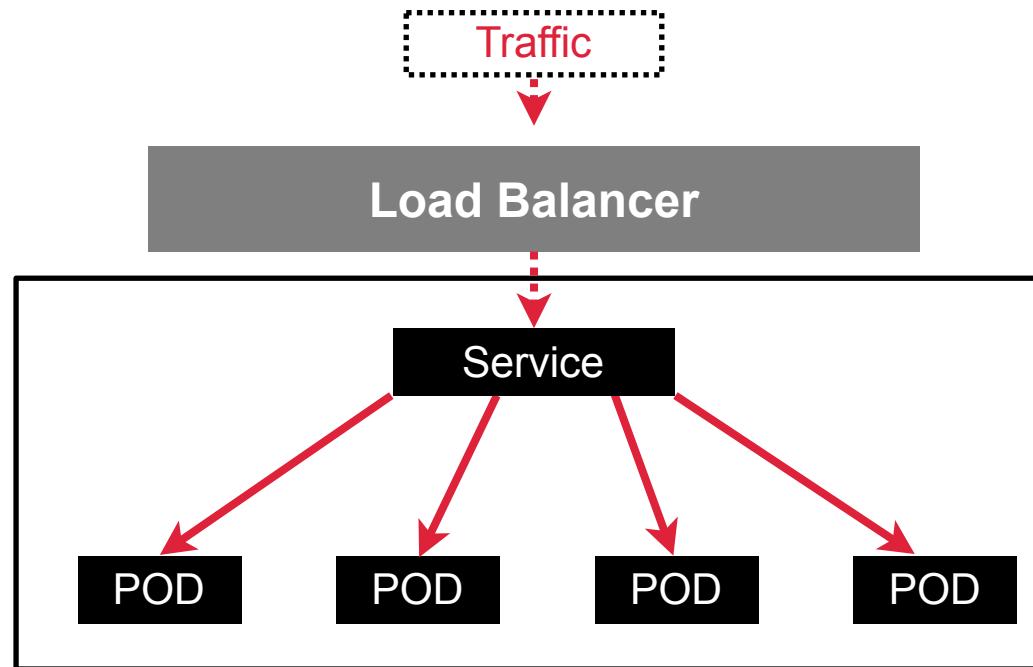
# NodePort



Opens up a designated port on each node that forwards traffic to the service - ideal for applications that need to be accessed outside the cluster like web apps or APIs



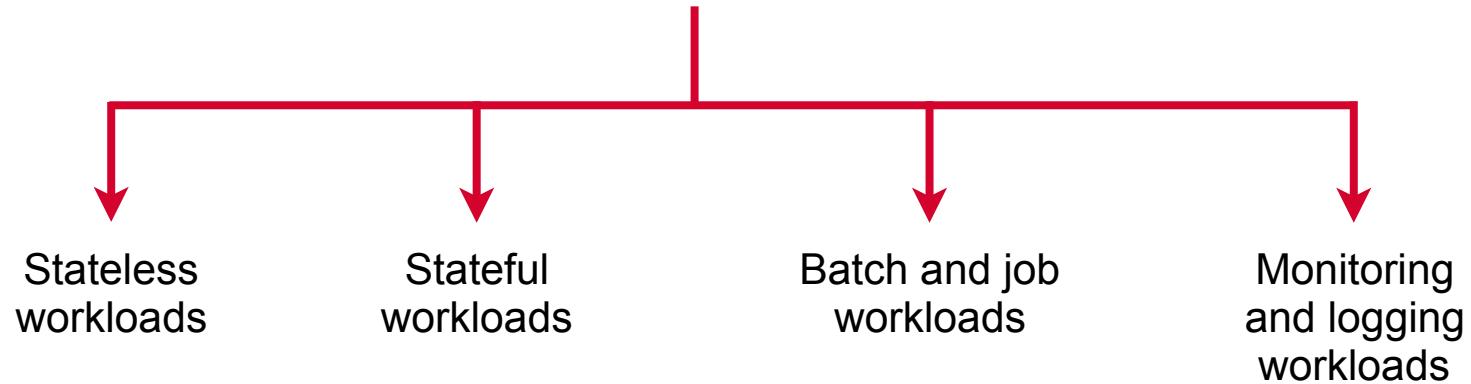
# LoadBalancer



Allows external connections to the service and distributes traffic to the different nodes running the service

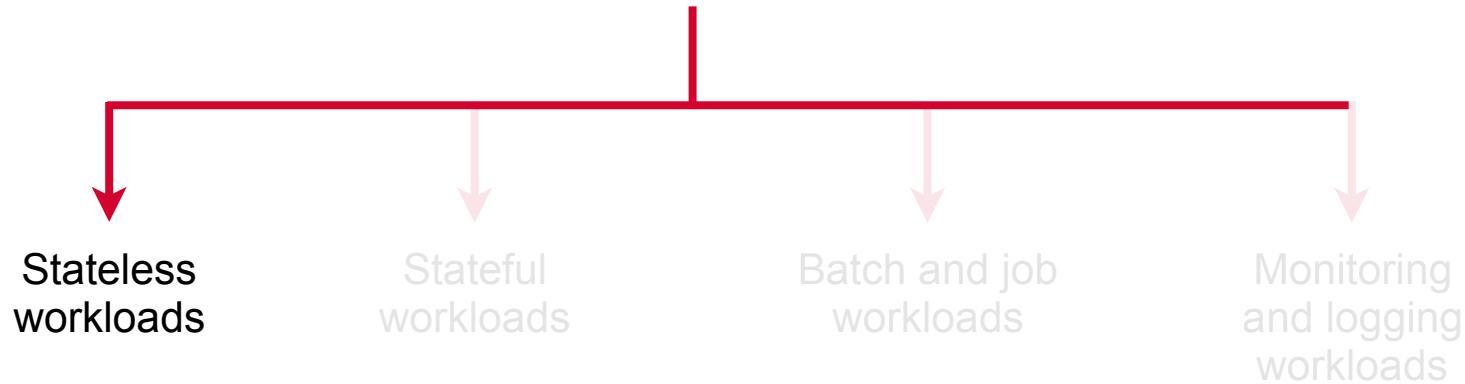


# Kubernetes Workloads





# Kubernetes Workloads





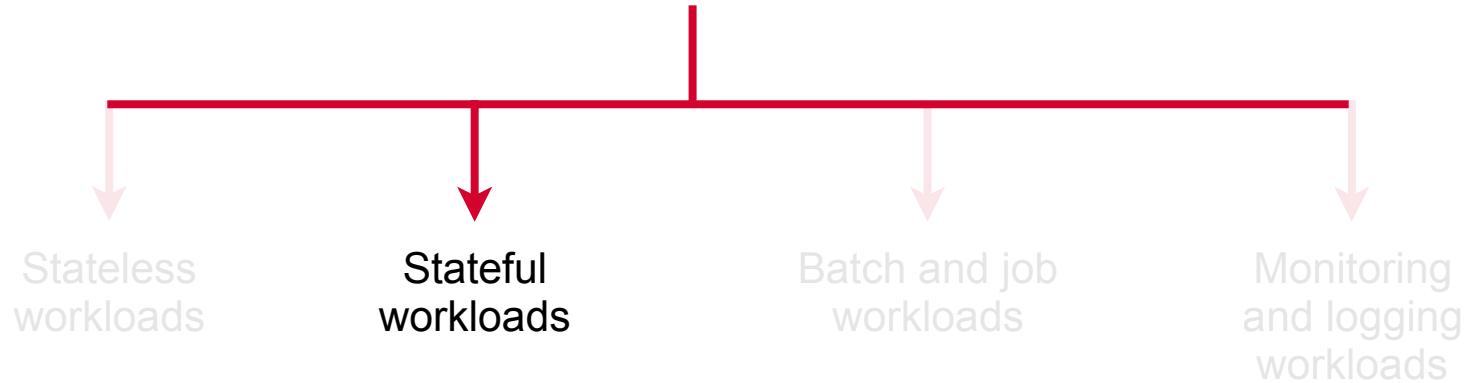
# Stateless Workloads

- Most commonly run using the **Deployment** object
  - Each pod operates independently
  - Does not retain data between sessions
- Web applications, microservices, and APIs that do not store data locally in the pod





# Kubernetes Workloads





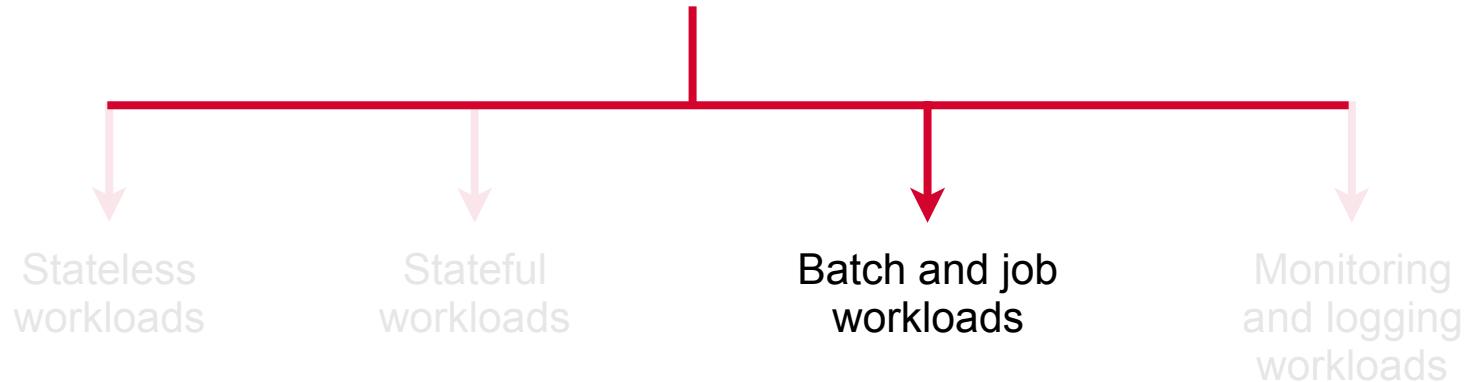
# Stateful Workloads

- **StatefulSet** object used to run stateful applications
  - Apps require persistent storage and unique identities for each pod
  - Databases, message queues, other applications that need stable storage and consistent naming





# Kubernetes Workloads





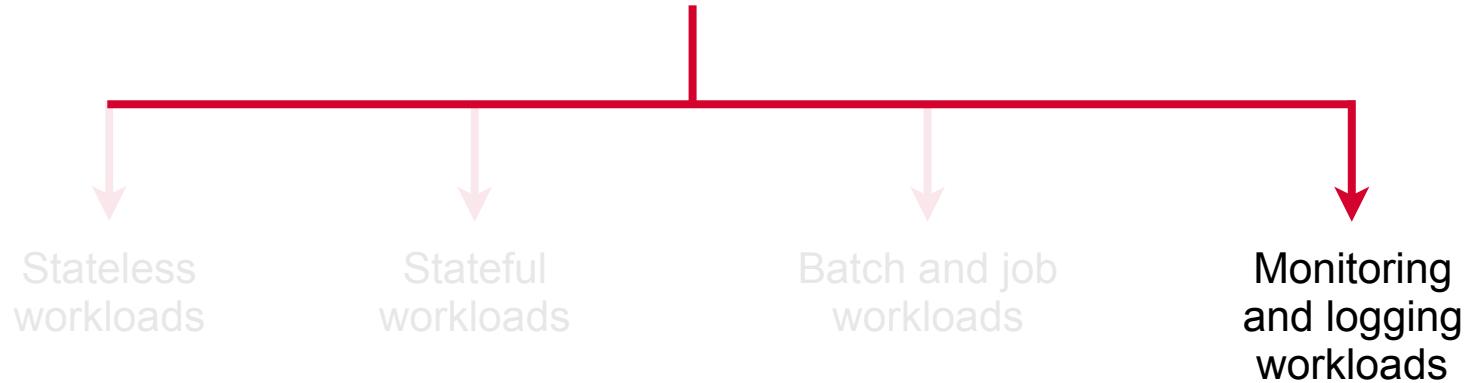
# Batch and Job Workloads

- The **Job** object managed one-off task or batch processing
  - A single job that needs to run to completion
- The **CronJob** is a specialized job that runs on a scheduled bases
  - Useful for tasks to be executed at regular intervals





# Kubernetes Workloads





# Monitoring and Logging Workloads

- A **DaemonSet** ensures that a copy of a pod is running on every node in the cluster
  - Used for system-level services running on every node
  - Logging agents, monitoring agents, or node-level proxies





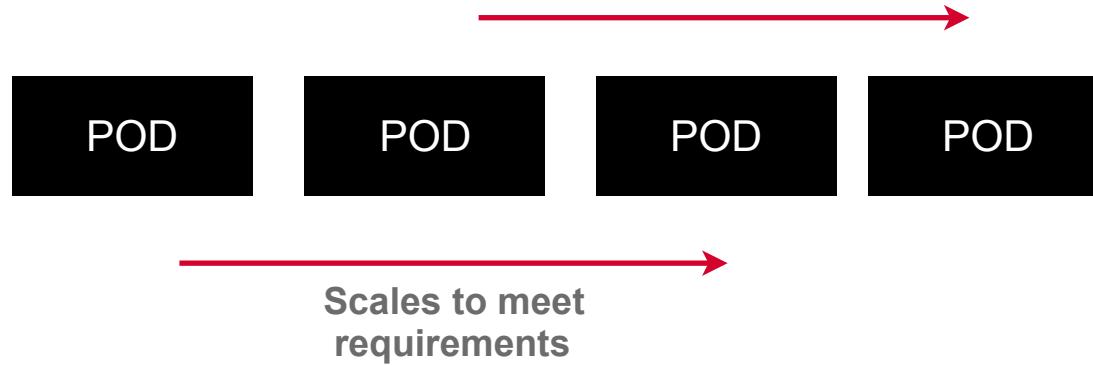
# Autoscaling

Horizontal Pod Autoscaler

Vertical Pod Autoscaler



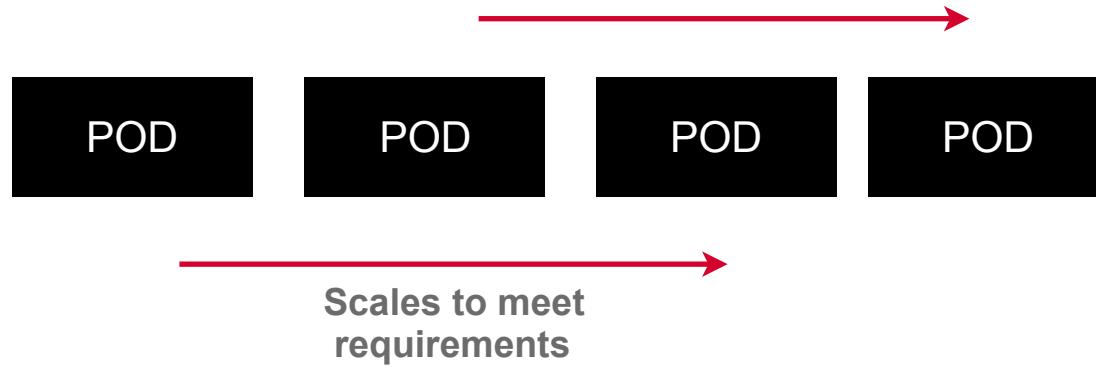
# Horizontal Pod Autoscaler



Adjusts the number of pods based on metrics such as CPU utilisation, memory usage, or other custom metrics



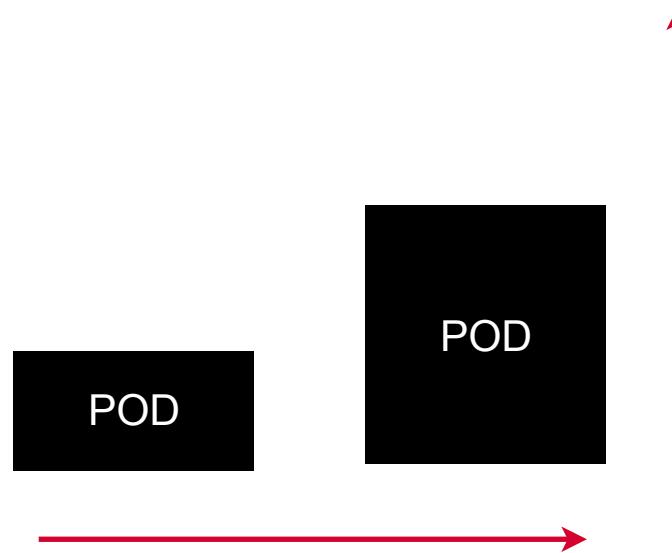
# Horizontal Pod Autoscaler



Monitors the performance metrics of pods and adds pods to match load - useful for stateless applications that handle fluctuating traffic



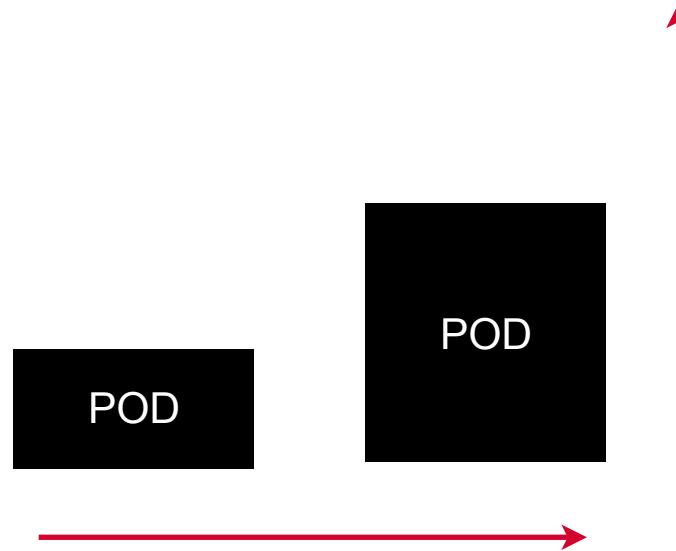
# Vertical Pod Autoscaler



Adjusts resource requests and CPU and memory limits of existing pods based on real-time usage



# Vertical Pod Autoscaler



Scales pods vertically by increasing or decreasing the resources over time - useful for stateful applications that need more compute or memory as the application scales



# Benefits of GKE

- Use GCP's load balancing for VMs
- Automatic scaling of nodes in cluster
- Automatic upgrades for software on nodes
- Node auto-repair for node health and availability
- Logging and monitoring using GCP's cloud monitoring





# GKE Mode of Operation

Autopilot Mode

Standard Mode



# Autopilot Mode

- More managed GKE experience
- GKE manages the underlying infrastructure - get a more serverless experience
- Node configuration, autoscaling, auto-upgrades, baseline security and networking configurations
- Implements best practices for security, scalability, and cost optimization by default





# Autopilot Mode

- **Cost effective:** Only pay for compute resources that your workloads use while running
- **Automation:** Google creates and manages nodes, scales nodes and workloads based on traffic
- **Security:** Enables many security settings and automatically applies security patches





# Standard Mode

- Complete control over all GKE configuration settings
- Manage configurations for node pools, security, scheduling, scaling, resource management, version management and software upgrades





## Use Standard Mode If:

- You want granular control over your configuration settings
- You want to install or modify software running on the nodes themselves i.e. change node OS
- Use certain features that are only available in the Standard Mode (GKE Sandbox, Cloud TPU)
- Test alpha features in open source Kubernetes





**Google recommends that you use an Autopilot cluster unless you have a specific need for a Standard cluster**



# Public and Private Clusters

Public clusters

Private clusters



# Public and Private Clusters

Public clusters

Private clusters

Nodes have public IP addresses and the cluster is accessible from the internet - needed when workloads must interact with external services



# Public and Private Clusters

Public clusters

Private clusters

Nodes have private IP addresses and the cluster is accessible only from your VPC - security and isolation are important and access to external services not required



## GKE Release Channels

Allow you to select a specific stream of Kubernetes versions that receive updates at a particular pace



# GKE Release Channels

Rapid Channel

Regular Channel

Stable Channel



# GKE Release Channels

Rapid Channel

Regular Channel

Stable Channel

Earliest access to new Kubernetes versions and features - ideal for development and testing environments. Likely to be frequent breaking changes



# GKE Release Channels

Rapid Channel

Regular Channel

Stable Channel

Balances new features with stability - receives updates and features less frequently than the rapid channel. Recommended for most production workloads



# GKE Release Channels

Rapid Channel

Regular Channel

Stable Channel

Offers the highest stability. Gets new features only when they have been thoroughly tested. Recommended for mission-critical applications

# GKE

Your company is running two different workloads on Google Kubernetes Engine (GKE). The first is a business-critical workload that requires high compute power and must be highly reliable. The second is a less critical workload that can run on general-purpose VMs and also take advantage of spot instances to minimize costs. How should you configure GKE to ensure optimal resource allocation and cost-efficiency for each workload?

- A. Use the same node pool for both workloads, but configure autoscaling for cost efficiency.
- B. Set up different node pools for each workload: one with high-compute VMs for the business-critical workload and another with general-purpose VMs and spot instances for the less critical workload.
- C. Use only general-purpose VMs for both workloads and enable preemptible VMs to reduce costs for the critical workload.
- D. Configure the business-critical workload to run on spot instances to lower the cost of high-compute resources.



# GKE

Your company is running two different workloads on Google Kubernetes Engine (GKE). The first is a business-critical workload that requires high compute power and must be highly reliable. The second is a less critical workload that can run on general-purpose VMs and also take advantage of spot instances to minimize costs. How should you configure GKE to ensure optimal resource allocation and cost-efficiency for each workload?

- A. Use the same node pool for both workloads, but configure autoscaling for cost efficiency.
- B. Set up different node pools for each workload: one with high-compute VMs for the business-critical workload and another with general-purpose VMs and spot instances for the less critical workload.**
- C. Use only general-purpose VMs for both workloads and enable preemptible VMs to reduce costs for the critical workload.
- D. Configure the business-critical workload to run on spot instances to lower the cost of high-compute resources.



# GKE

You are deploying a workload on Google Kubernetes Engine (GKE) and want access to the latest features while following Google-recommended best practices for setting up your GKE cluster. How should you configure your GKE environment?

- A. Use GKE Standard mode with manual upgrades to control the update process and get the latest features when you choose.
- B. Use GKE Autopilot and enroll in the rapid release channel to automatically access the latest features while following best practices for cluster management.
- C. Use GKE Standard mode with the static version setting to avoid unexpected changes and ensure stability.
- D. Use GKE Autopilot with the regular release channel for stable updates and manage the updates manually to ensure the latest features are applied.



# GKE

You are deploying a workload on Google Kubernetes Engine (GKE) and want access to the latest features while following Google-recommended best practices for setting up your GKE cluster. How should you configure your GKE environment?

- A. Use GKE Standard mode with manual upgrades to control the update process and get the latest features when you choose.
- B. Use GKE Autopilot and enroll in the rapid release channel to automatically access the latest features while following best practices for cluster management.**
- C. Use GKE Standard mode with the static version setting to avoid unexpected changes and ensure stability.
- D. Use GKE Autopilot with the regular release channel for stable updates and manage the updates manually to ensure the latest features are applied.



# GKE

Your company is deploying a sensitive application on Google Kubernetes Engine (GKE) that must meet strict security and compliance requirements. The application needs to run in a secure environment with minimal exposure to the public internet while leveraging automation to manage cluster operations. What should you do to meet these requirements?

- A. Set up a public GKE Autopilot cluster and configure firewall rules to limit access.
- B. Use a GKE Standard cluster with private nodes and manually manage cluster operations.
- C. Set up a private Autopilot cluster to ensure the cluster is secured, isolated from the public internet, and managed automatically.
- D. Deploy a GKE Standard cluster with public access and regularly update security patches manually.



# GKE

Your company is deploying a sensitive application on Google Kubernetes Engine (GKE) that must meet strict security and compliance requirements. The application needs to run in a secure environment with minimal exposure to the public internet while leveraging automation to manage cluster operations. What should you do to meet these requirements?

- A. Set up a public GKE Autopilot cluster and configure firewall rules to limit access.
- B. Use a GKE Standard cluster with private nodes and manually manage cluster operations.
- C. Set up a private Autopilot cluster to ensure the cluster is secured, isolated from the public internet, and managed automatically.**
- D. Deploy a GKE Standard cluster with public access and regularly update security patches manually.



# GKE

Your company is running a web application on Google Kubernetes Engine (GKE) that experiences fluctuating traffic patterns. You want to ensure the application remains highly available during peak loads by automatically adjusting the number of pods. At the same time, you want to receive recommendations on optimizing resource usage for each pod without automatically changing resource allocations. What should you do?

- A. Manually adjust the pod count during peak traffic times and monitor resource usage.
- B. Configure the Horizontal Pod Autoscaler for availability, and configure the Vertical Pod Autoscaler recommendations for suggestions.
- C. Set up only the Vertical Pod Autoscaler to automatically adjust pod resource allocations based on traffic.
- D. Use a fixed number of pods and enable the Cluster Autoscaler to manage resource allocation.



# GKE

Your company is running a web application on Google Kubernetes Engine (GKE) that experiences fluctuating traffic patterns. You want to ensure the application remains highly available during peak loads by automatically adjusting the number of pods. At the same time, you want to receive recommendations on optimizing resource usage for each pod without automatically changing resource allocations. What should you do?

- A. Manually adjust the pod count during peak traffic times and monitor resource usage.
- B. Configure the Horizontal Pod Autoscaler for availability, and configure the Vertical Pod Autoscaler recommendations for suggestions.**
- C. Set up only the Vertical Pod Autoscaler to automatically adjust pod resource allocations based on traffic.
- D. Use a fixed number of pods and enable the Cluster Autoscaler to manage resource allocation.



# GKE

You are planning to deploy a containerized application to Google Cloud using a Kubernetes manifest. You want to maintain full control over the deployment process but reduce the effort needed to configure and manage the underlying infrastructure. What approach should you take?

- A. Deploy the application using GKE Autopilot to minimize infrastructure management while maintaining control over the Kubernetes deployment.
- B. Use Google Cloud Run to deploy the container and let it manage the Kubernetes resources for you.
- C. Set up a GKE Standard cluster, manually manage the nodes, and deploy your Kubernetes manifest.
- D. Use Compute Engine to deploy the container and manage all the infrastructure manually.



# GKE

You are planning to deploy a containerized application to Google Cloud using a Kubernetes manifest. You want to maintain full control over the deployment process but reduce the effort needed to configure and manage the underlying infrastructure. What approach should you take?

- A. Deploy the application using GKE Autopilot to minimize infrastructure management while maintaining control over the Kubernetes deployment.
- B. Use Google Cloud Run to deploy the container and let it manage the Kubernetes resources for you.
- C. Set up a GKE Standard cluster, manually manage the nodes, and deploy your Kubernetes manifest.
- D. Use Compute Engine to deploy the container and manage all the infrastructure manually.

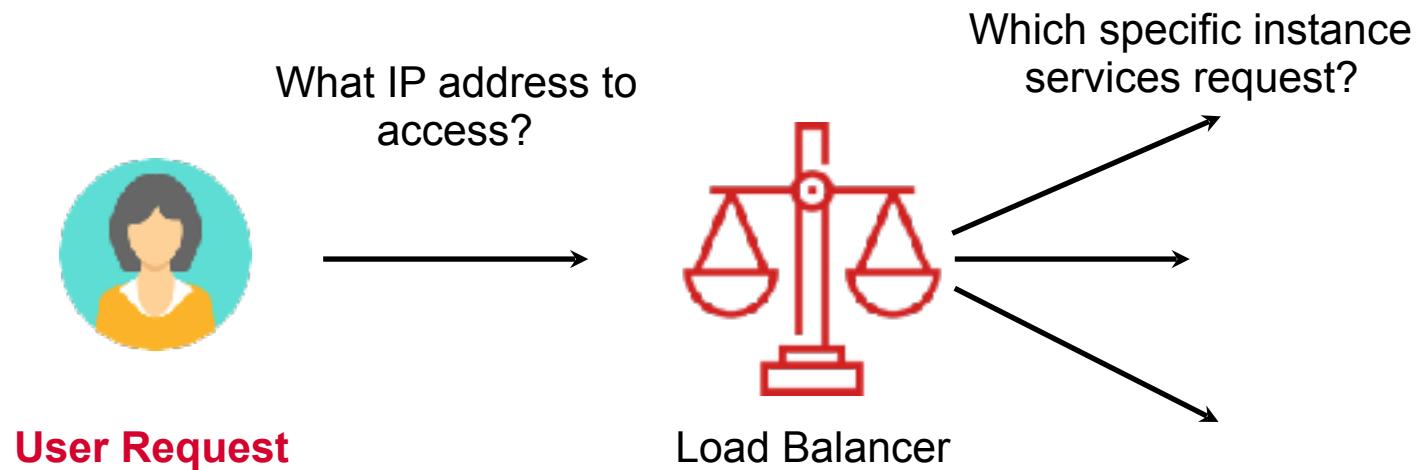


# Load Balancing





# Load Balancers

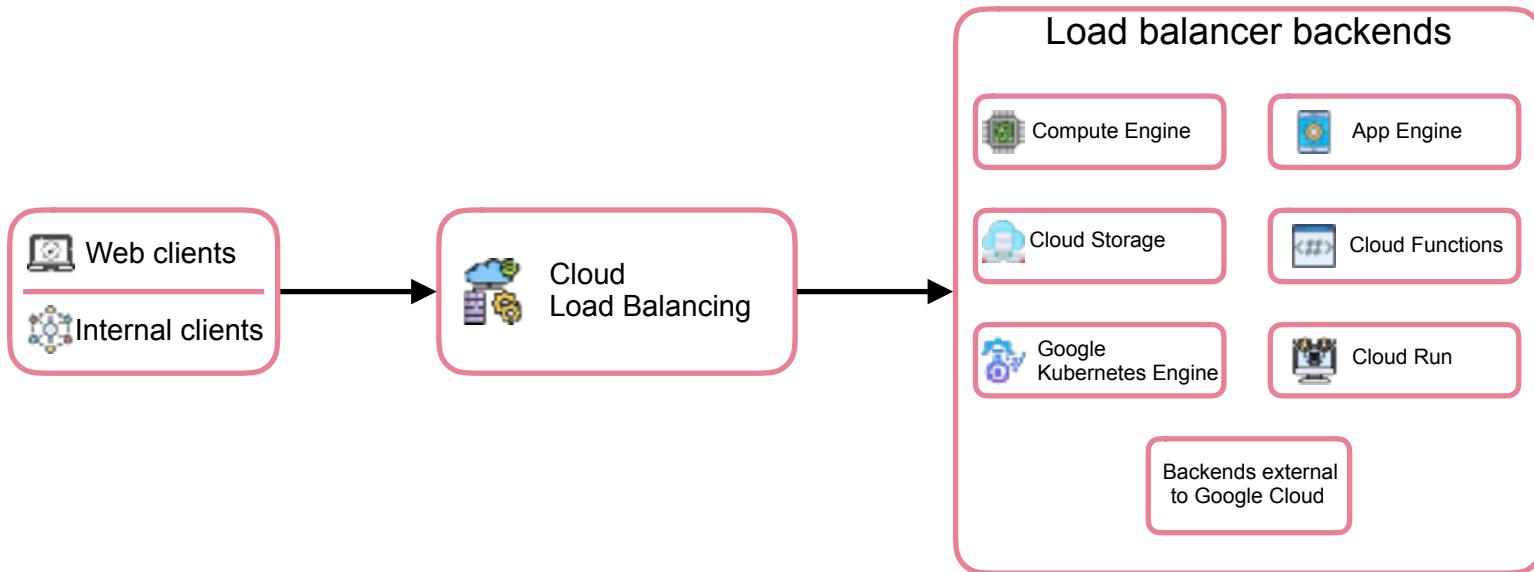


Backend  
Service





# Load Balancers Used with Multiple Backends





# Load Balancers

- Complex service
- Many moving parts
- Basic idea
  - Stable front-end IP
  - Forwarding rules to funnel traffic
  - Connect to backend service
  - Distribute load intelligently
  - Health checks to avoid unhealthy instances





Load balancers **distribute** traffic to  
resources close to users and meet  
**high-availability** requirements



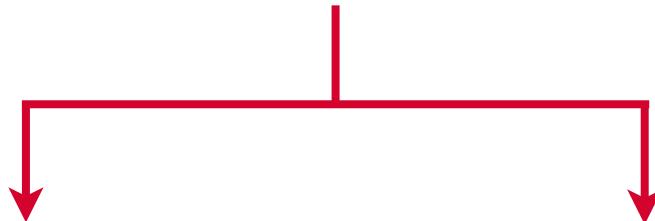
# Load Balancers on the GCP

- Fully managed, software-defined, redundant and highly available
- Supports > 1 million queries per second with high performance and low latency
- Autoscaling to meet increased traffic





# Load Balancing Categorization



## Global

Use when your users and instances are globally distributed, load balanced resources lie across multiple regions

## Regional

Use when instances and users are concentrated in one region



# Load Balancing Categorization





# 7 Layer OSI Network Stack

Routing decisions based on attributes of the request i.e. HTTP headers and the URL

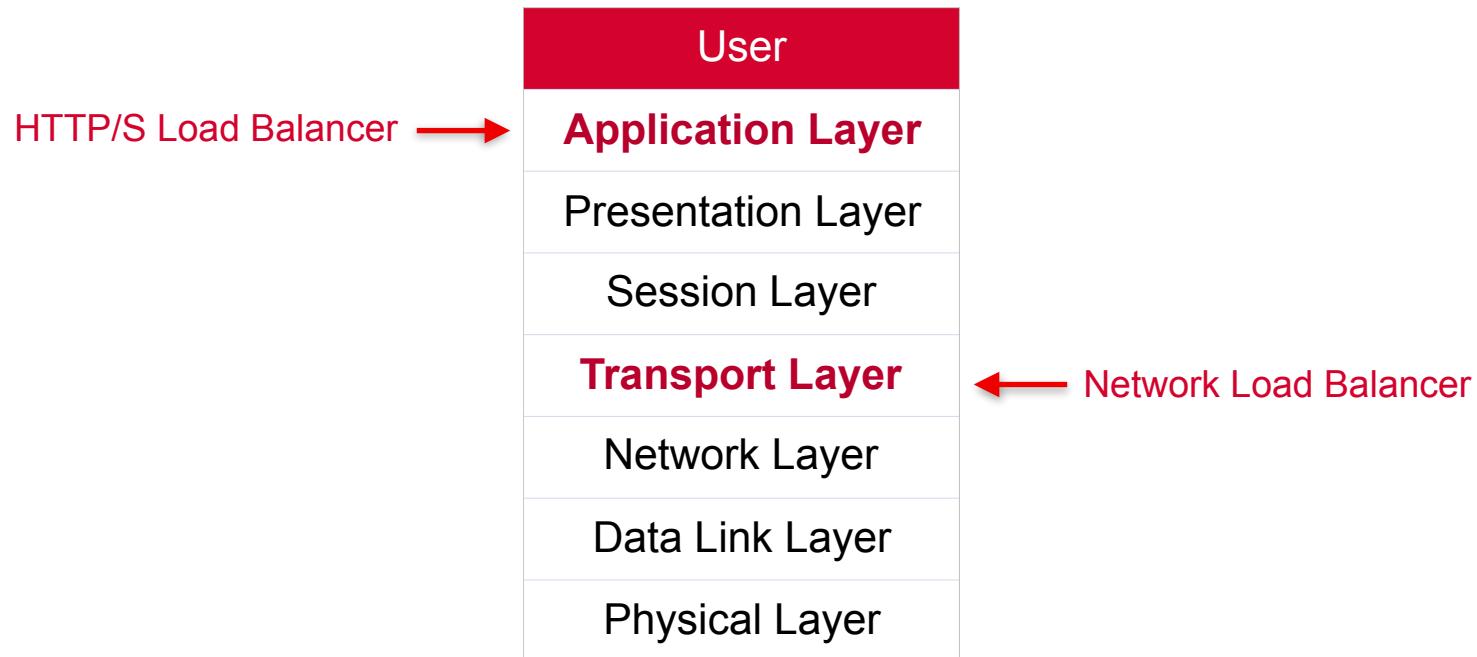


Direct traffic based on data from network and transport layer protocols such as TCP, UDP, ESP, GRE, ICMP, and ICMPv6





# 7 Layer OSI Network Stack





# Two Types of Load Balancers

Application Load  
Balancers

Network Load  
Balancers



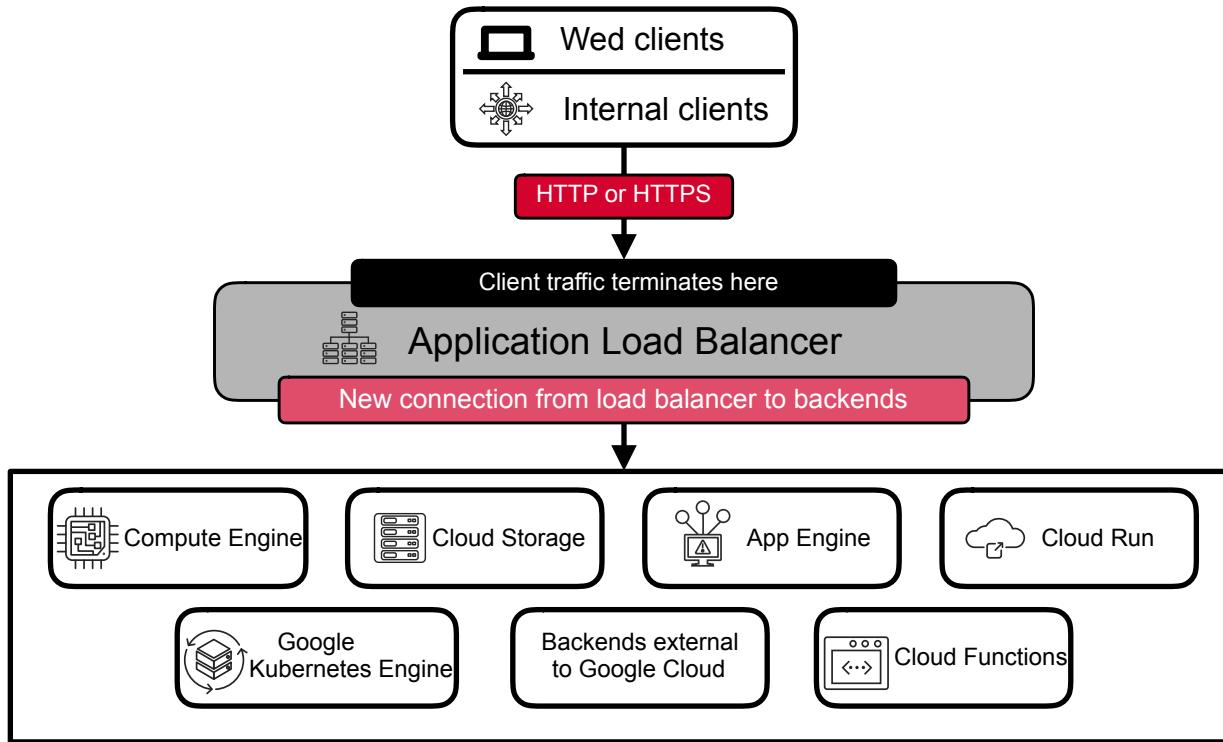
# Application Load Balancers

- **Proxy-based** layer 7 load balancers
- Allow you to scale your services behind a single IP
- Distributes HTTP and HTTPS traffic to Google backends and external backends
  - Compute Engine, GKE, Cloud Run



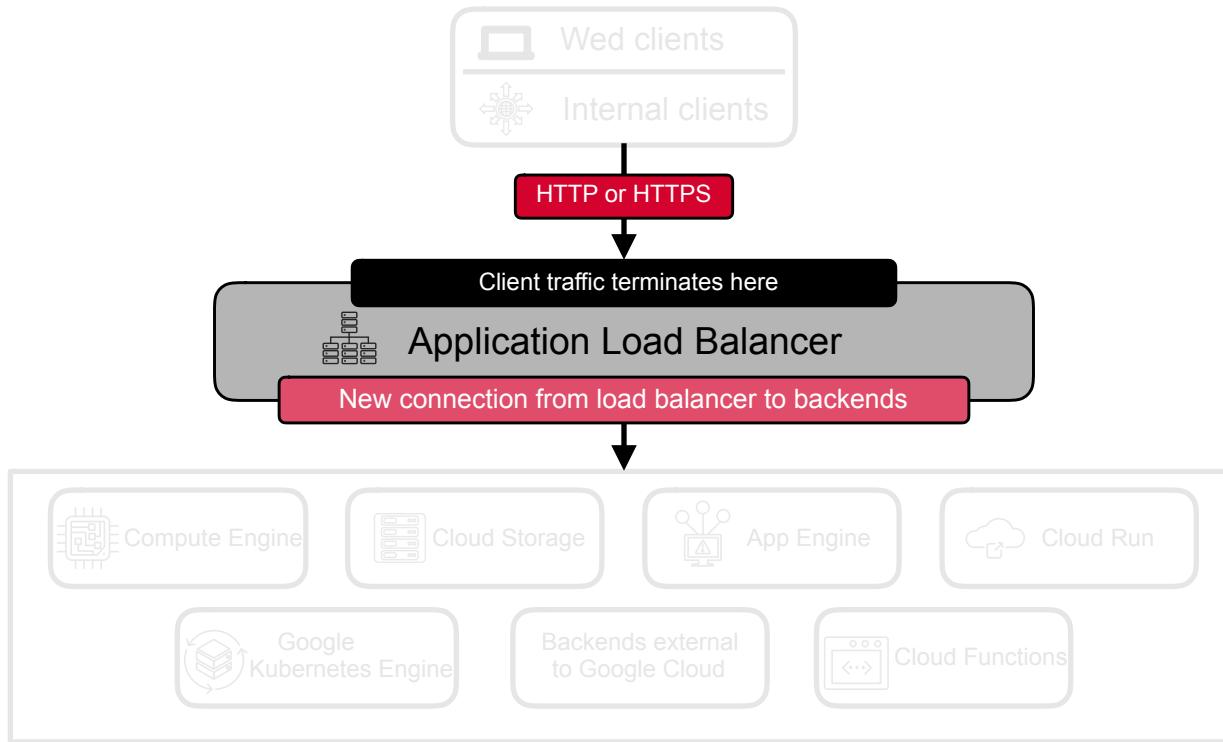


# Application Load Balancers





# Proxy Load Balancing





# URL-based Routing

The **HTTP(S) Load Balancer** can split traffic based on content using **URL-based routing rules**

The load balancer inspects the incoming request's URL path or hostname and direct the traffic to different backend services or instances based on predefined conditions



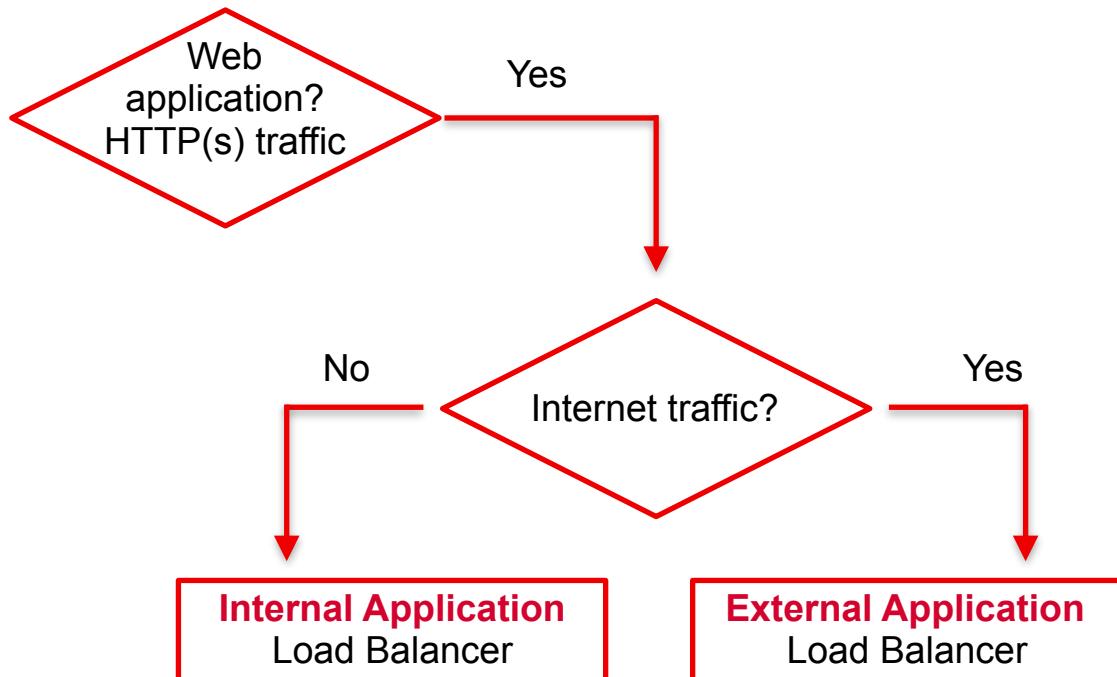
# URL-based Routing

- Path-based routing
  - example.com/images/\* routed to Backend Service 1
  - example.com/videos/\* routed to Backend Service 2
- Host-based routing
  - app.example.com routed to Backend Service 3
  - blog.example.com routed to Backend Service 4



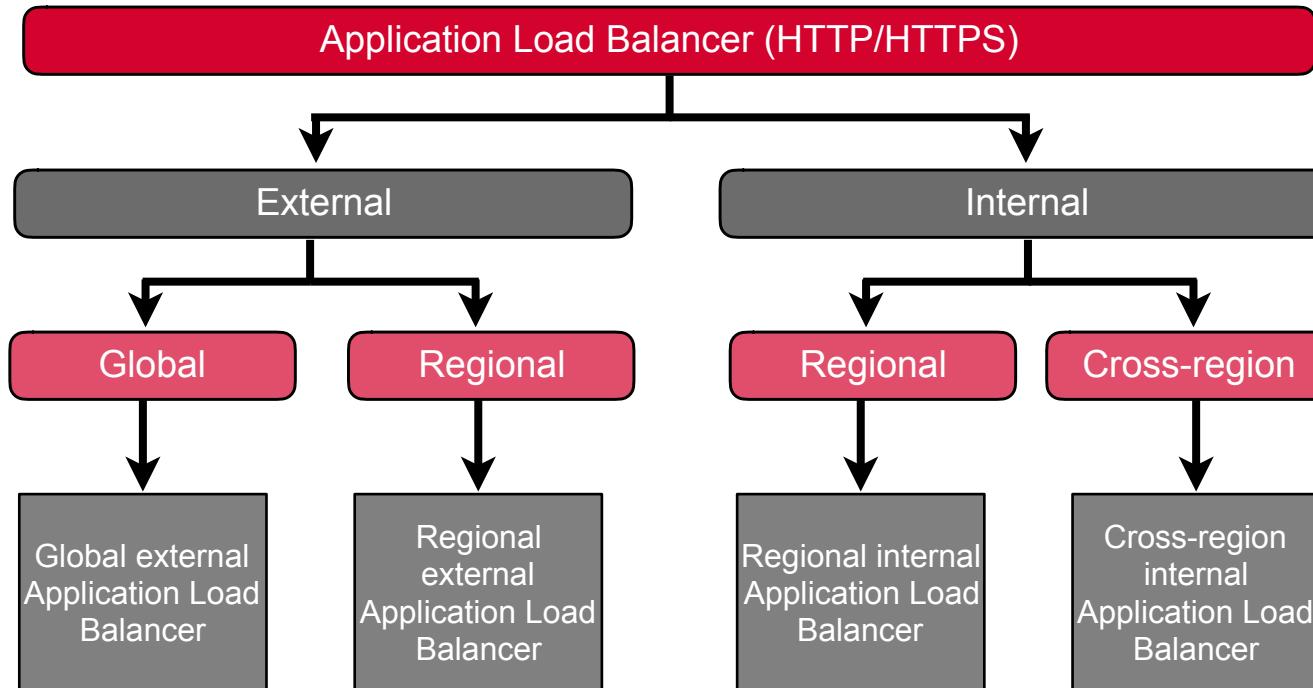


# Choosing Load Balancers





# Application Load Balancers





# Network Load Balancers

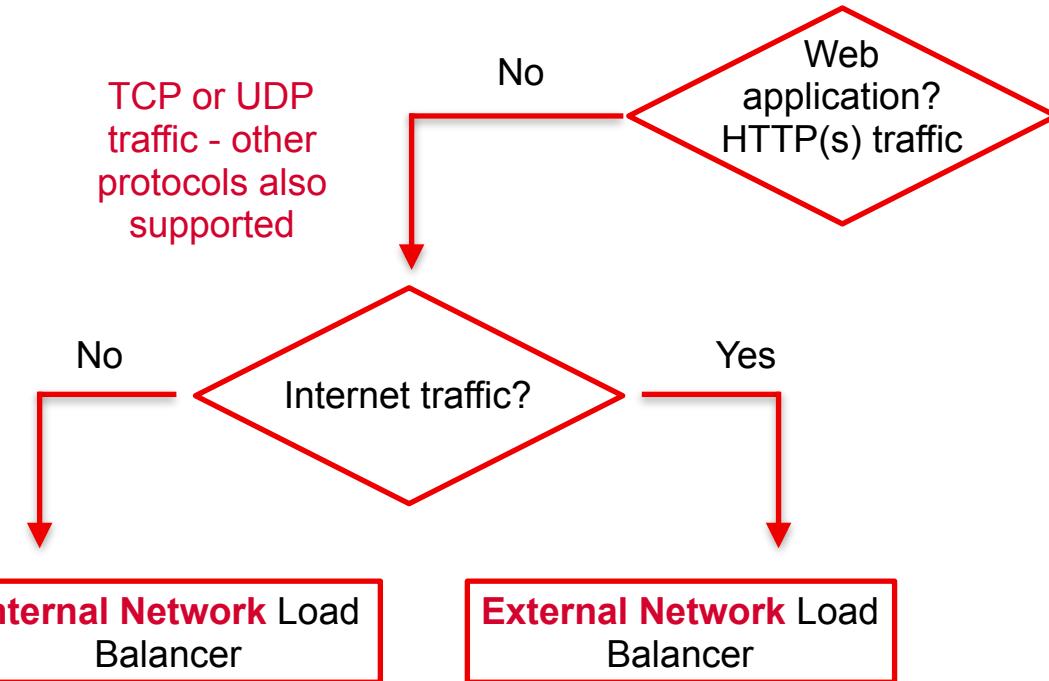
- Layer 4 load balancers
- Handle TCP, UDP, or other IP protocol traffic
- Can be of two types
  - Proxy
  - Passthrough





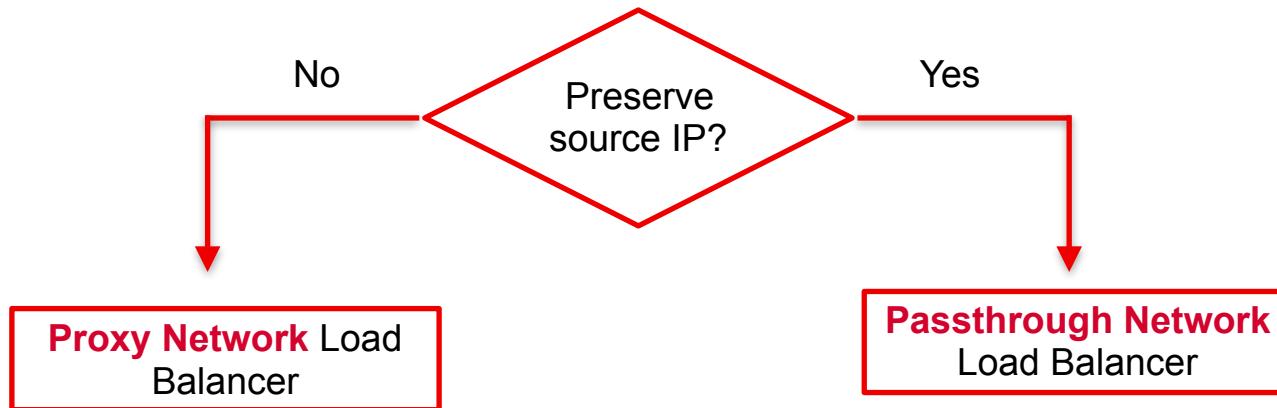
# Choosing Load Balancers

TCP or UDP traffic - other protocols also supported



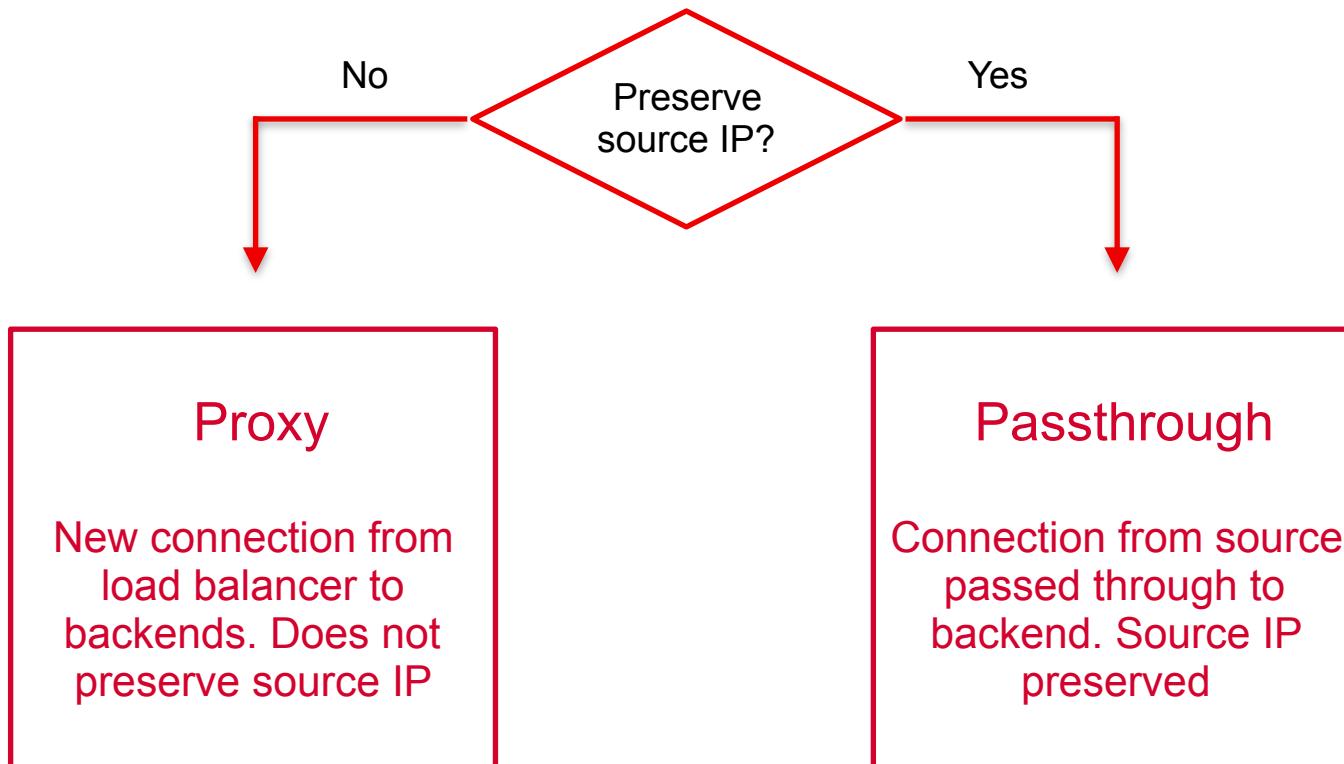


# Choosing Network Load Balancers



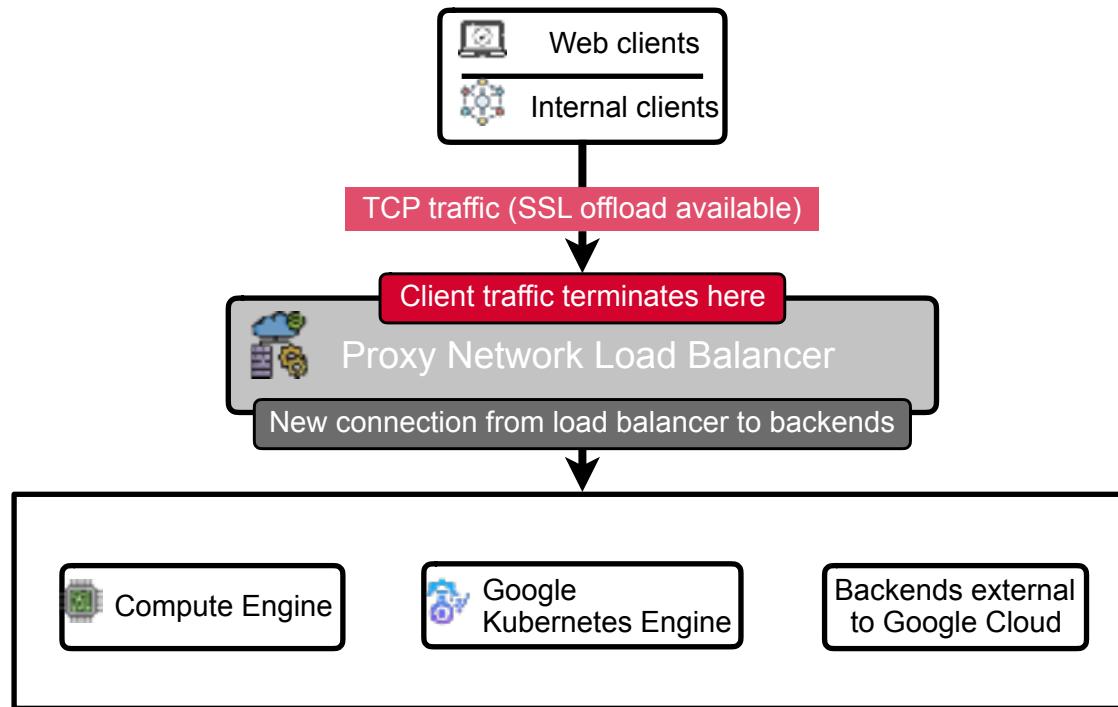


# Choosing Network Load Balancers



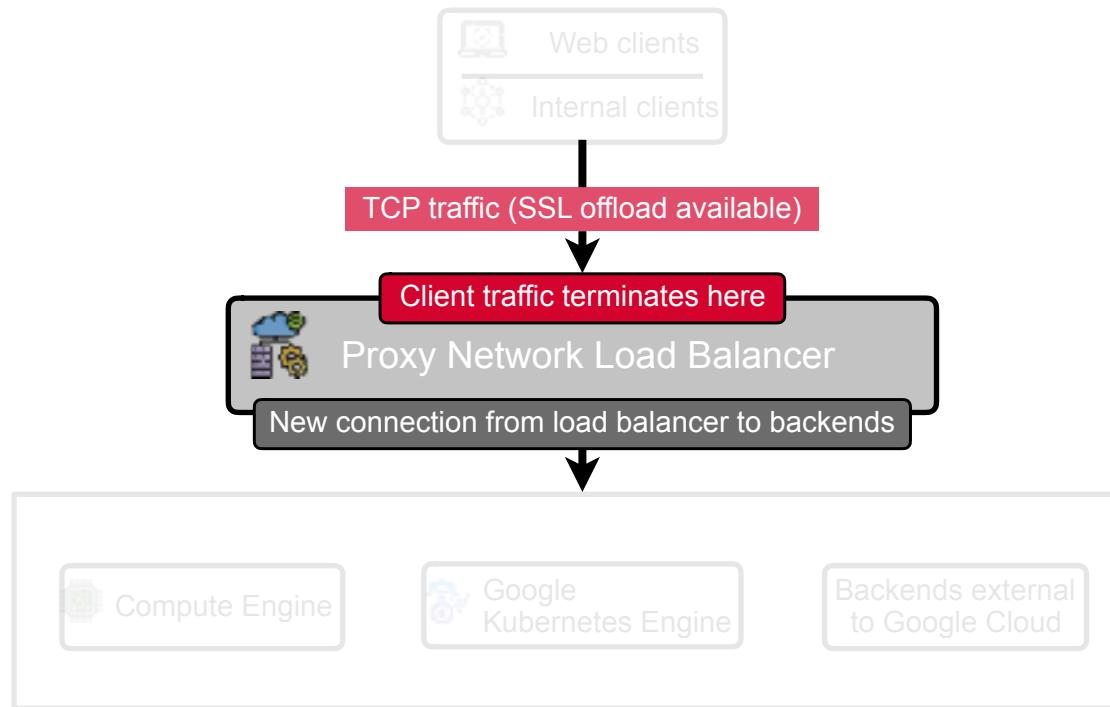


# Network Proxy Load Balancers





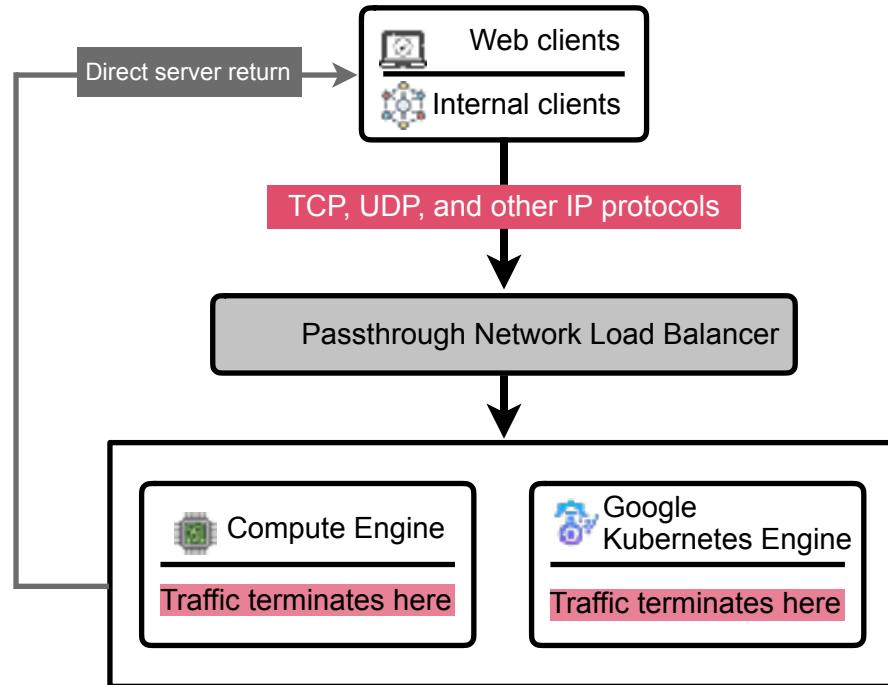
# Network Proxy Load Balancers – SSL Offload



The load balancer terminates the secure SSL/TLS connection, decrypts the data, and forwards the plain HTTP traffic to the backend servers.

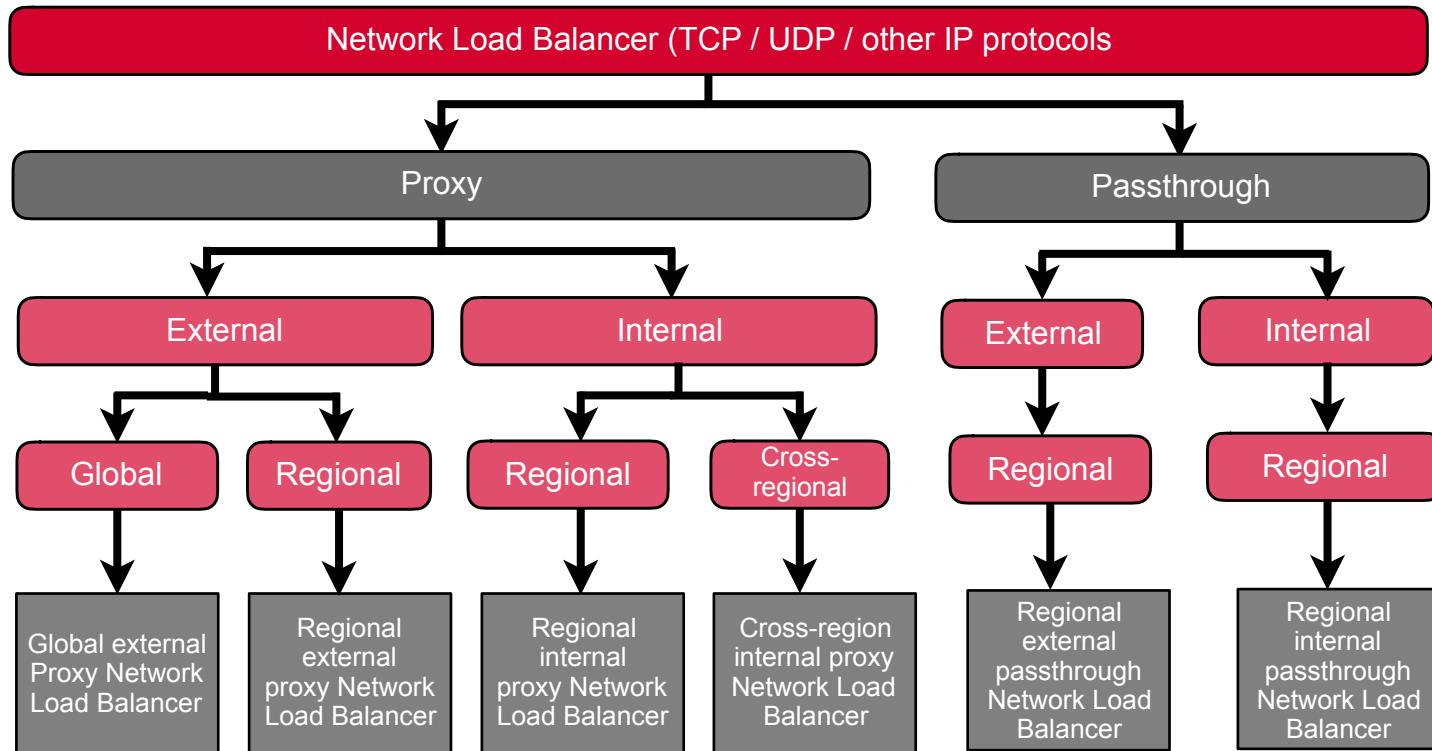


# Network Passthrough Load Balancers





# Network Load Balancers



---

# Load Balancers

Your company has deployed a network of IoT sensors around the world that send real-time data via UDP packets to Google Cloud for processing. The backend infrastructure consists of multiple virtual machines (VMs) to handle the incoming data stream and scale as needed. The data needs to be sent to a single IP address and load balanced. How would you accomplish this?

- A. Set up an Internal HTTP(s) load balancer to route the UDP traffic to the VMs.
- B. Deploy an external Network load balancer with SSL offload to handle packets
- C. Configure an external Network load balancer to manage UDP traffic
- D. Implement an internal TCP/UDP load balancer to handle the UDP packets



---

# Load Balancers

Your company has deployed a network of IoT sensors around the world that send real-time data via UDP packets to Google Cloud for processing. The backend infrastructure consists of multiple virtual machines (VMs) to handle the incoming data stream and scale as needed. The data needs to be sent to a single IP address and load balanced. How would you accomplish this?

- A. Set up an Internal HTTP(s) load balancer to route the UDP traffic to the VMs.
- B. Deploy an external Network load balancer with SSL offload to handle packets
- C. Configure an external Network load balancer to manage UDP traffic**
- D. Implement an internal TCP/UDP load balancer to handle the UDP packets



# Load Balancers

Your company has set up a private microservices architecture on Google Cloud in a single region, where multiple backend services need to communicate with each other using internal IP addresses. One of the key requirements is to preserve the source IP of the client when traffic is routed between services. You are tasked with configuring a load balancer that meets these requirements for handling TCP traffic. What should you do?

- A. Configure an external TCP Proxy load balancer
- B. Set up an internal HTTP(s) load balancer with session affinity enabled.
- C. Implement an internal network passthrough load balancer
- D. Deploy an external network load balancer with TCP forwarding rules for internal traffic.



# Load Balancers

Your company has set up a private microservices architecture on Google Cloud in a single region, where multiple backend services need to communicate with each other using internal IP addresses. One of the key requirements is to preserve the source IP of the client when traffic is routed between services. You are tasked with configuring a load balancer that meets these requirements for handling TCP traffic. What should you do?

- A. Configure an external TCP Proxy load balancer
- B. Set up an internal HTTP(s) load balancer with session affinity enabled.
- C. Implement an internal network passthrough load balancer**
- D. Deploy an external network load balancer with TCP forwarding rules for internal traffic.



# Load Balancers

Your company has launched a new web application on Google Cloud that needs to serve HTTP traffic to users over the internet. The application is hosted on multiple virtual machines (VMs) to handle increasing traffic and ensure high availability. You want to expose the application using a single IP address and ensure that HTTP traffic from the internet is efficiently distributed across the VMs. What should you do?

- A. Set up an internal TCP/UDP load balancer to route traffic to the VMs
- B. Configure an external network load balancer to route traffic to the VMs
- C. Implement an external application load balancer to route traffic to the VMs
- D. Deploy an SSL proxy load balancer to route traffic to the VMs



# Load Balancers

Your company has launched a new web application on Google Cloud that needs to serve HTTP traffic to users over the internet. The application is hosted on multiple virtual machines (VMs) to handle increasing traffic and ensure high availability. You want to expose the application using a single IP address and ensure that HTTP traffic from the internet is efficiently distributed across the VMs. What should you do?

- A. Set up an internal TCP/UDP load balancer to route traffic to the VMs
- B. Configure an external network load balancer to route traffic to the VMs
- C. Implement an external application load balancer to route traffic to the VMs**
- D. Deploy an SSL proxy load balancer to route traffic to the VMs



O'REILLY®

# Identity and Access Management





# Cloud IAM

Manage identity and access control by defining **who** (identity) has **what access** (role) for **which resource**.



# Cloud IAM

Permission to access a resource is not granted directly to the end user. Instead, permissions are grouped into **roles**, and roles are granted to authenticated **principals**.



# Cloud IAM

Permission to access a resource is not granted directly to the end user. Instead, permissions are grouped into roles, and roles are granted to authenticated principals.

- **Principal:** GCP identity - user, group, service account
- **Role:** Collection of permissions
- **Policy:** Binding members to a role



# Role-based Access Control



Identity



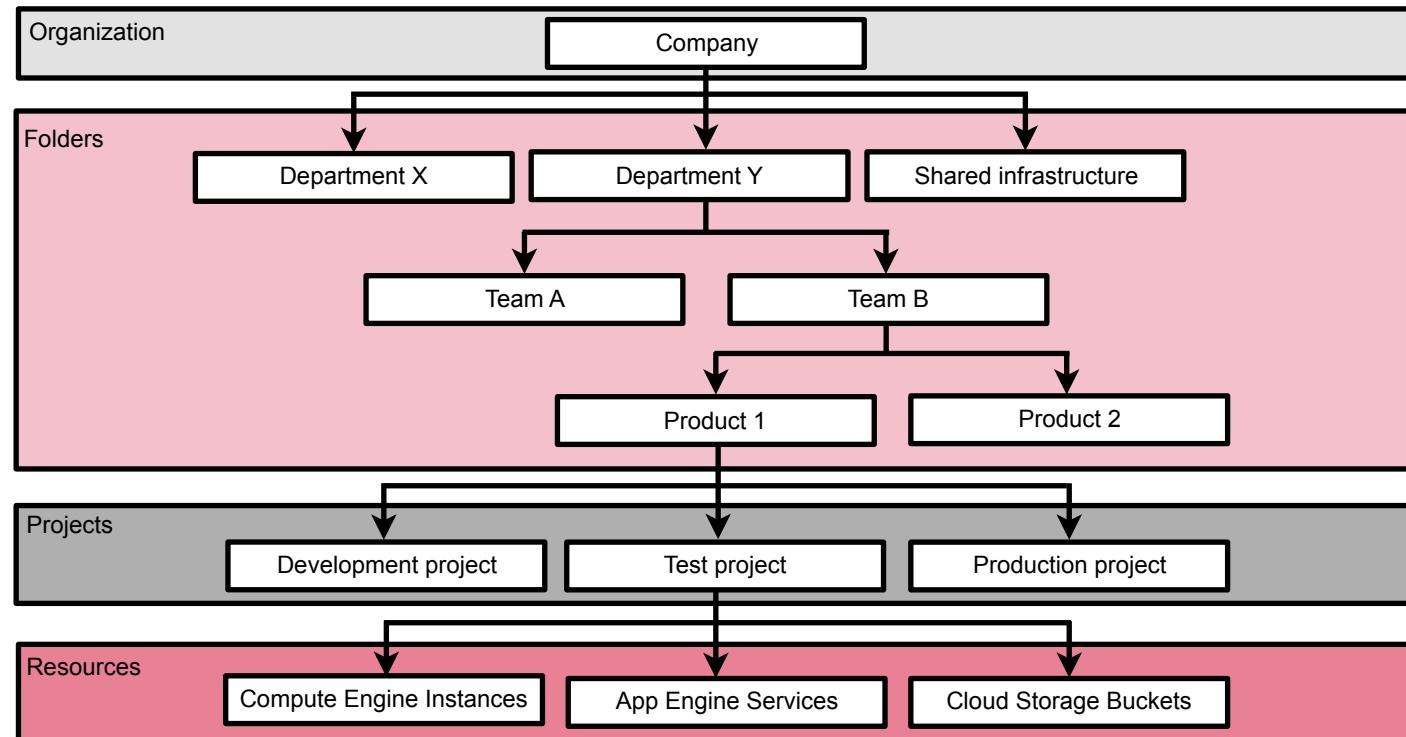
Permissions



Resource

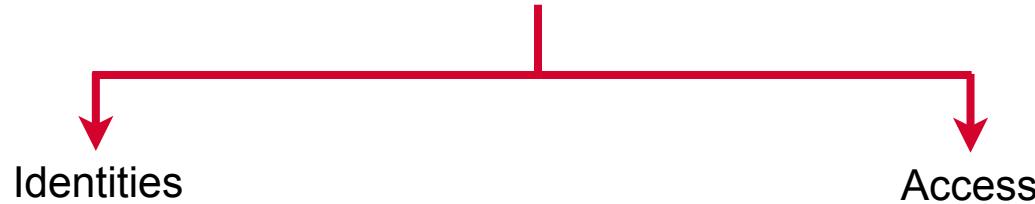


# Google Cloud Hierarchy



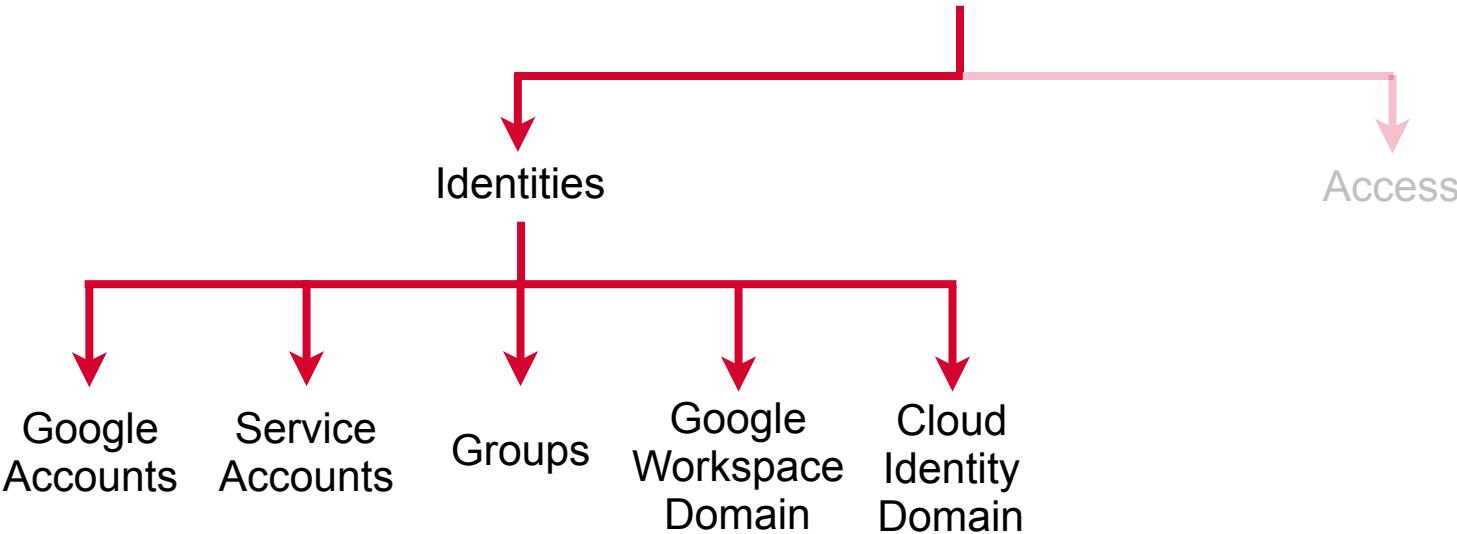


# Identity and Access Management (IAM)





# Identity and Access Management (IAM)





# Google Accounts

A Google account represents a developer, an administrator, or any other person who interacts with the Google Cloud.



# Service Accounts

A service account is an account that belongs to **your application** instead of to an individual end user.



# Google Groups

A Google Group is a named **collection of Google accounts and service accounts**. Every group has a unique email address that is associated with the group.



# Google Workspace Domains

A Google Workspace domain represents a **virtual group of all the Google accounts** that have been created in an organization's account.

Google Workspace domains represent your organization's Internet domain name.



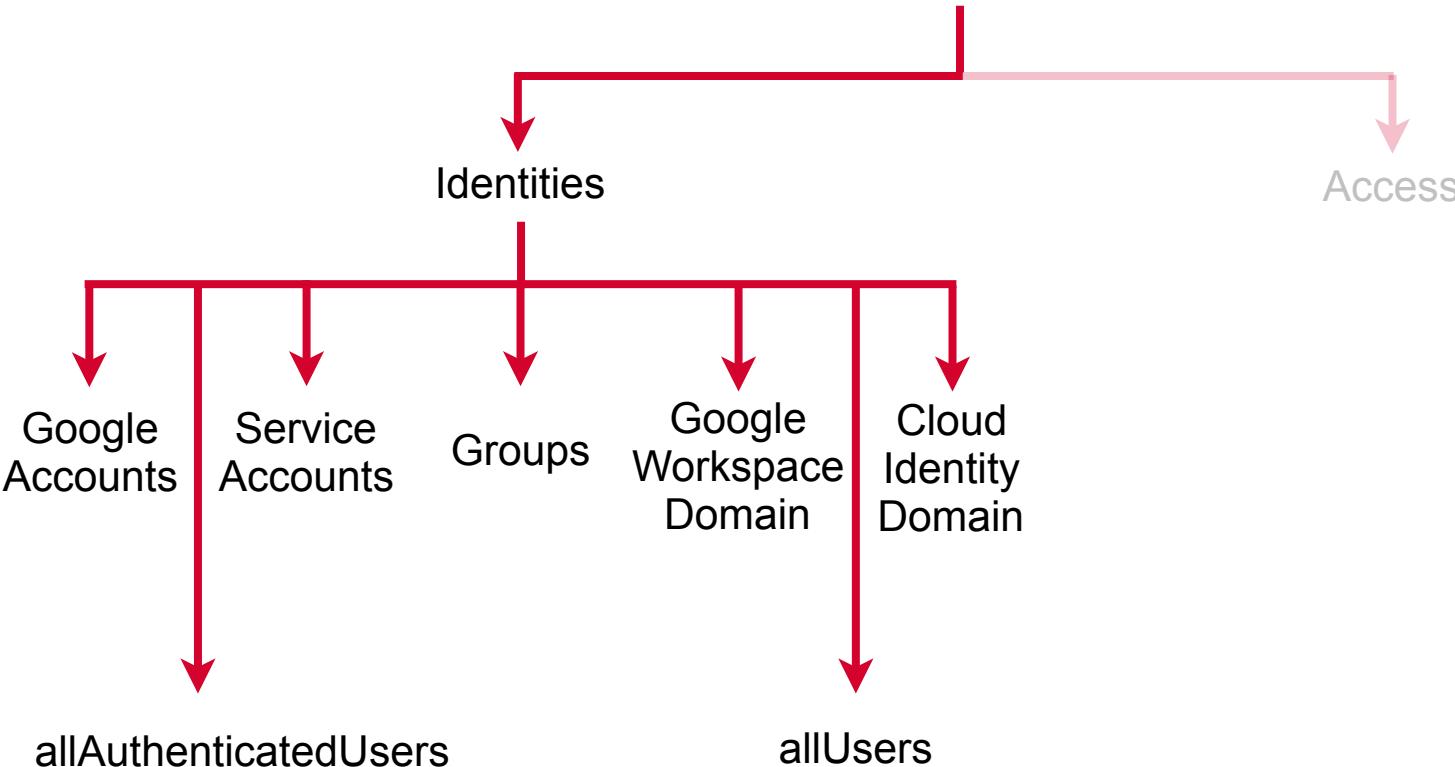
# Cloud Identity Domains

A Cloud Identity domain is like a Google Workspace domain because it represents a virtual group of all Google accounts in an organization.

However, Cloud Identity domain users don't have access to Google Workspace applications and features.



# Identity and Access Management (IAM)





# allAuthenticatedUsers

Special identifier that represents all service accounts and all users on the internet who have authenticated with a Google Account.

- Accounts need not be connected to a Google Workspace account or Cloud Identity domain e.g. Gmail accounts
- Unauthenticated users i.e. anonymous users NOT included

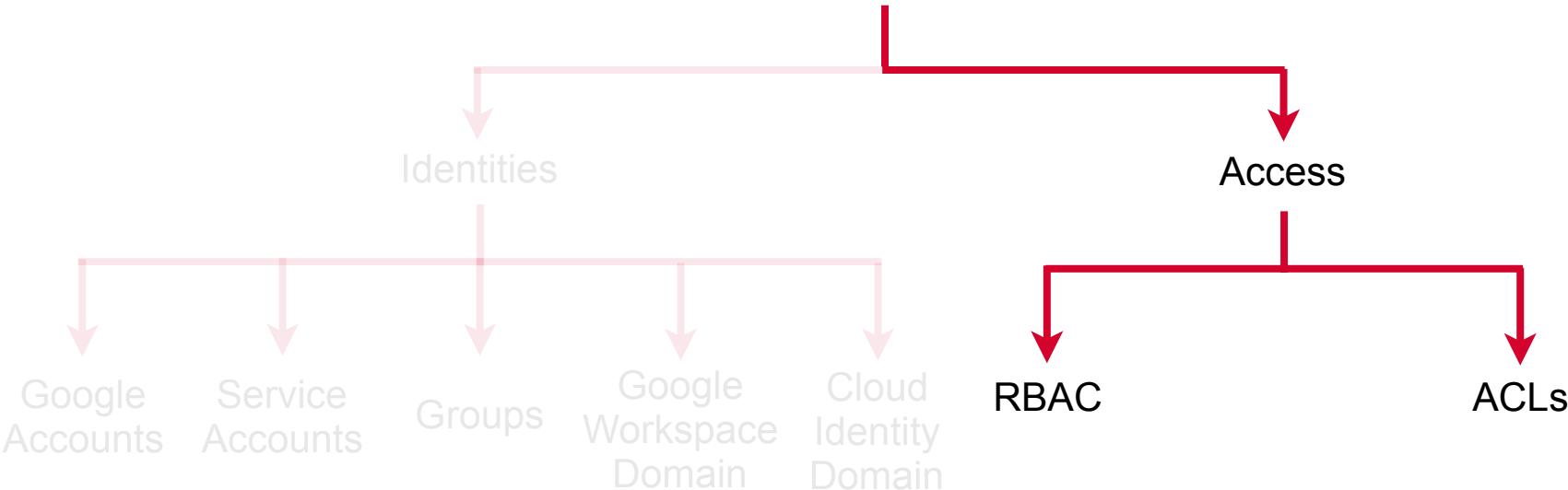


## allUsers

Special identifier that represents anyone who is on the internet, including authenticated and unauthenticated users.

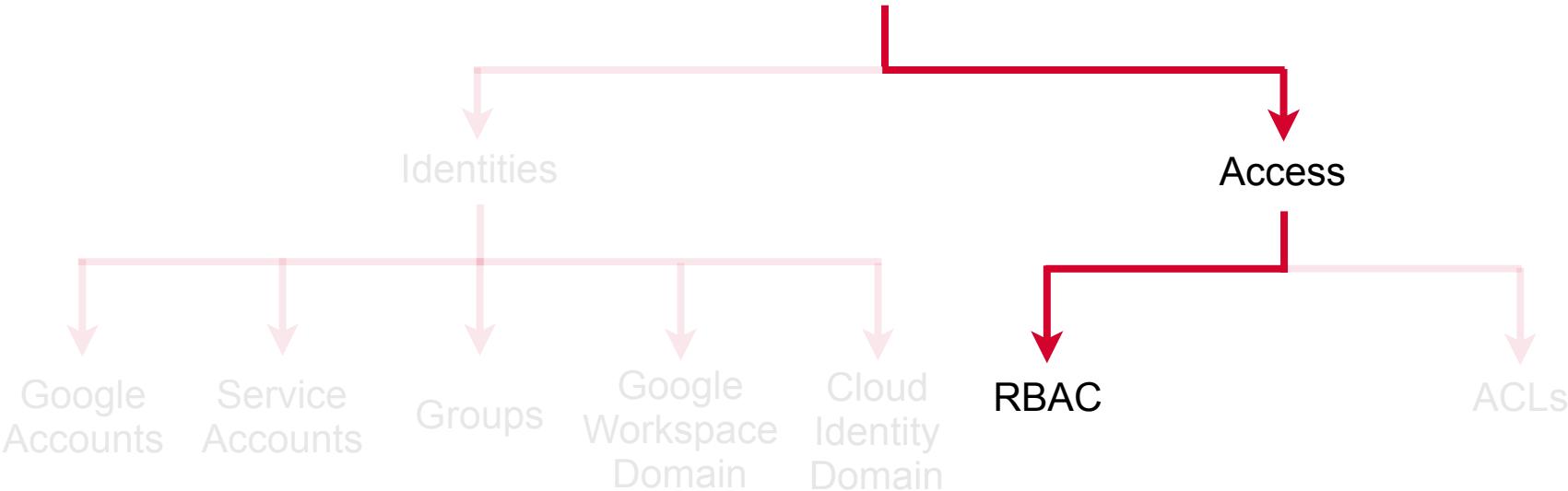


# Identity and Access Management (IAM)





# Identity and Access Management (IAM)





# A Role is Just a Collection of Permissions

Role-compute.instanceAdmin

## Permissions

compute.instances.delete

compute.instances.get

compute.instances.list

compute.instances.  
setMachineType

compute.instances.start

compute.instances.stop

...



# Structure of Each Permission “service.resource.verb”

**storage.buckets.create**: Allows the user to create new Cloud Storage buckets.

**compute.instances.start**: Grants permission to start existing virtual machine instances in Google Compute Engine

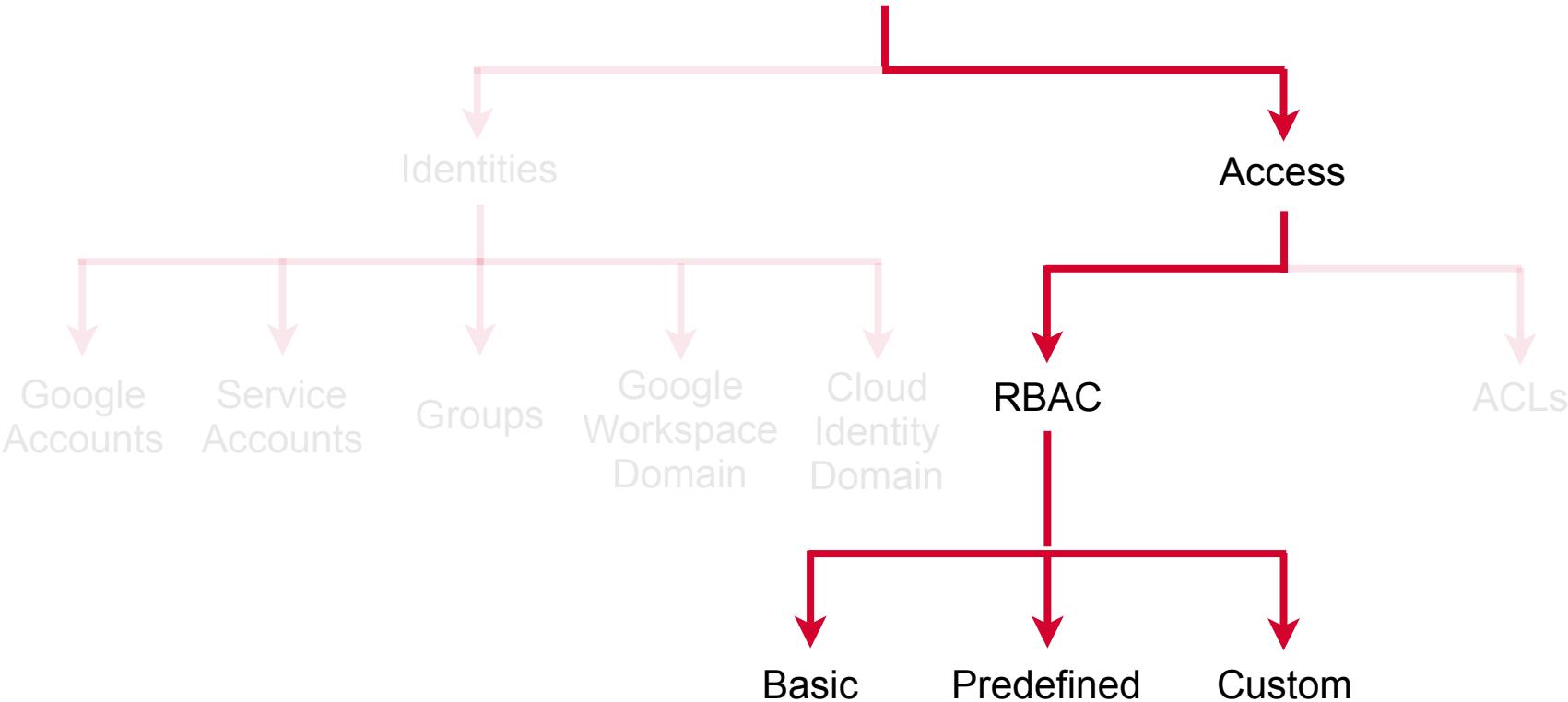
**bigrquery.datasets.get**: Provides read access to metadata of BigQuery datasets without allowing access to the actual data.

**pubsub.topics.publish**: Enables the user to publish messages to a Pub/Sub topic.





# Identity and Access Management (IAM)





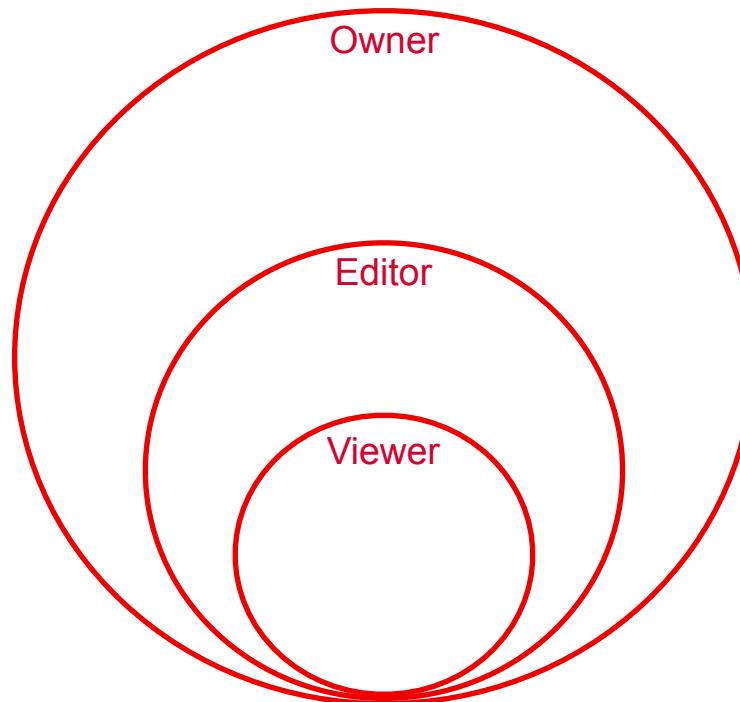
## Basic Roles

Three concentric roles that existed prior to the introduction of Cloud IAM:  
Owner, Editor, and Viewer of any resource.

- Includes 1000s of permissions across all Google Cloud Services
- Historically available before Cloud IAM was a complete feature.
- **DO NOT USE - unless there is no alternative**



# Owner, Editor, Viewer Roles are Concentric



If you grant both the broader and limited role (such as Editor and the Viewer) to the same person, only the broader role is granted to them.



# Predefined Roles

- Project Roles
- App Engine Roles
- BigQuery Roles
- Cloud Bigtable Roles
- Cloud Billing Roles





# Predefined Roles

## **roles/bigquery.dataViewer**

```
bigquery.datasets.get  
bigquery.datasets.getIamPolicy  
bigquery.models.getData  
bigquery.models.getMetadata  
bigquery.models.list  
bigquery.routines.get  
bigquery.routines.list  
bigquery.tables.export  
bigquery.tables.get  
bigquery.tables.getData  
bigquery.tables.list  
resourcemanager.projects.get  
resourcemanager.projects.list
```





**Predefined roles are always preferred  
over Basic and Custom roles**



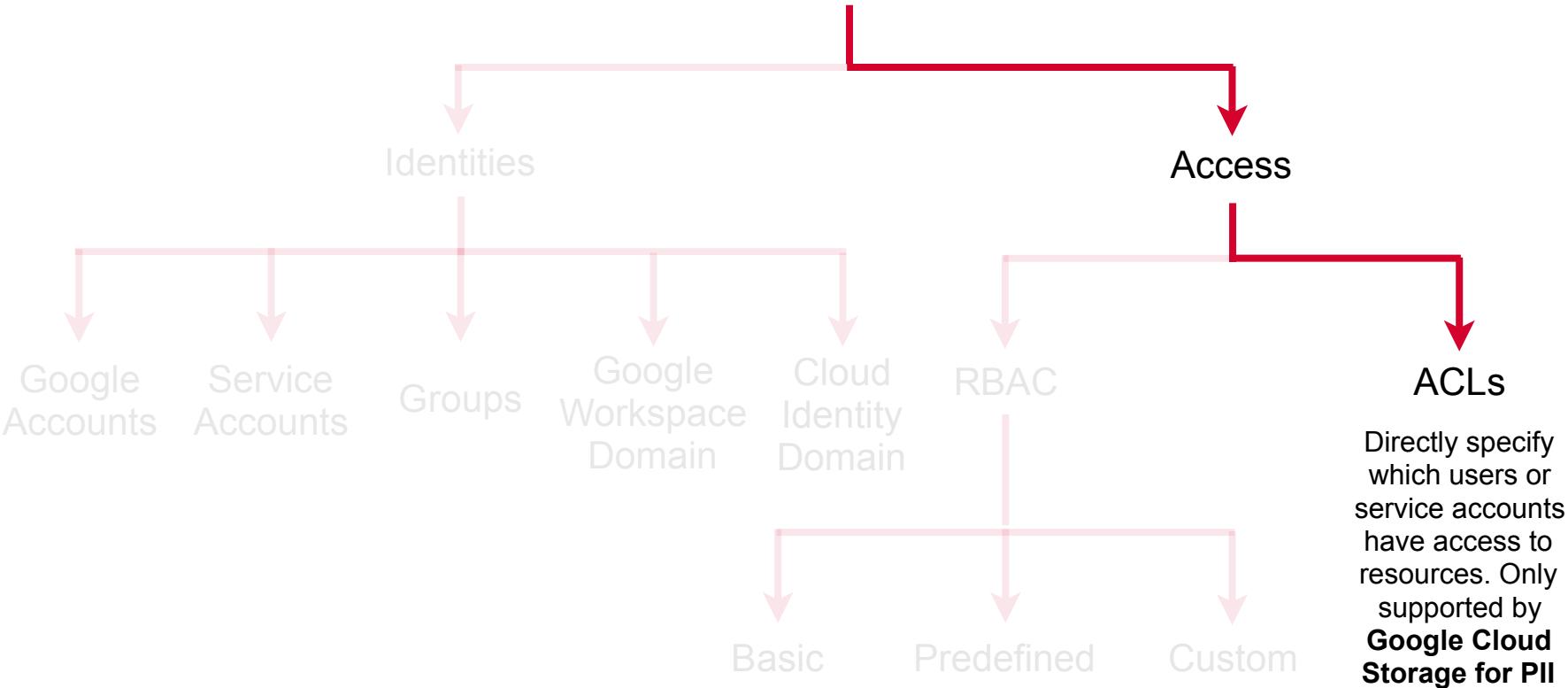
## Custom Roles

User-defined roles that bundle one or more supported permissions tailored to meet your specific needs.

Not maintained by Google; when new permissions, features, or services are added to GCP, your custom roles will not be updated automatically.

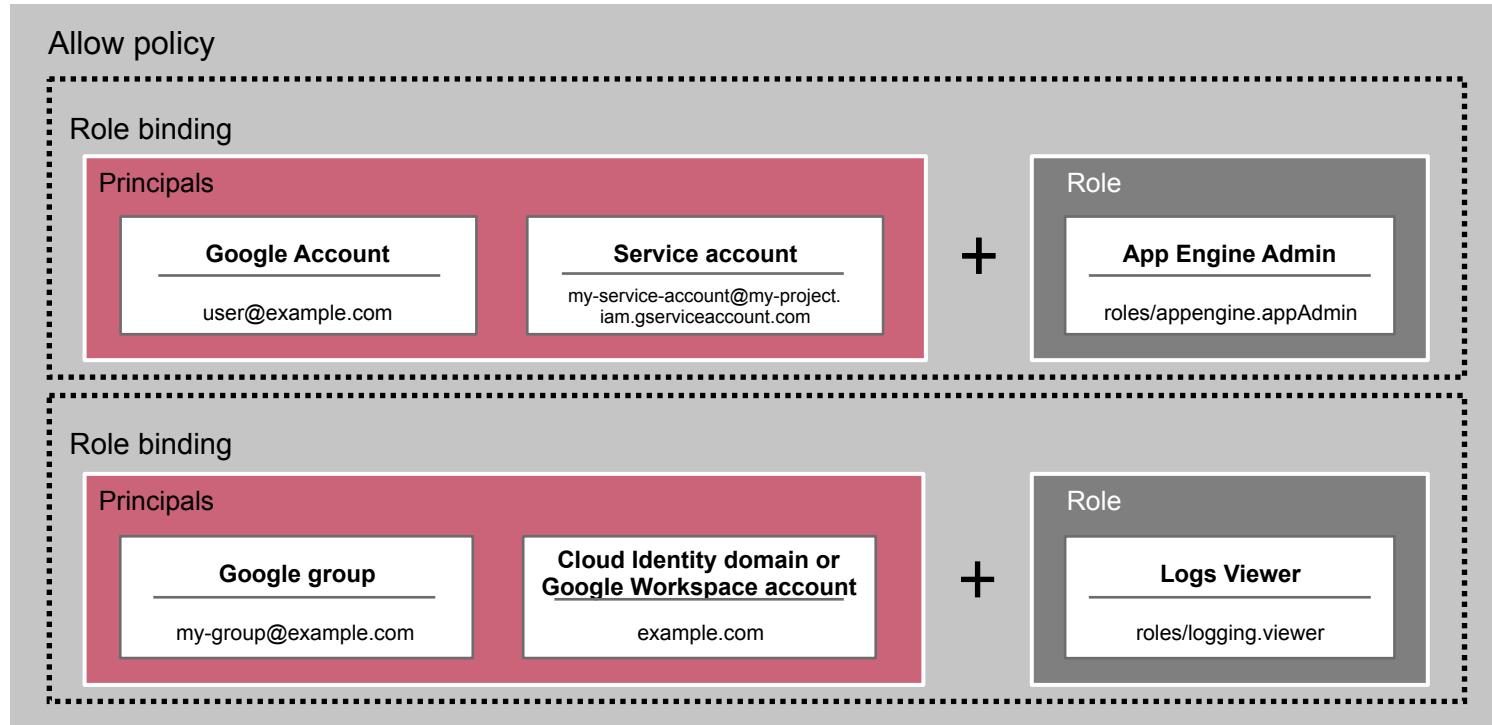


# ACLs Not Part of the IAM Service on Google





# Allow Policy Used to Grant Roles to Users



An *allow policy* defines and enforces what roles are granted to which principals.



# Deny Policy to Restrict Access to Resources

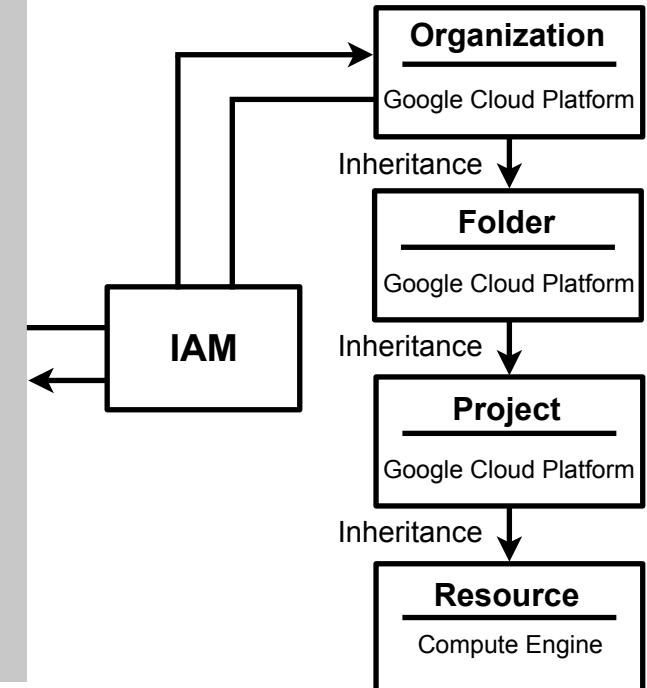
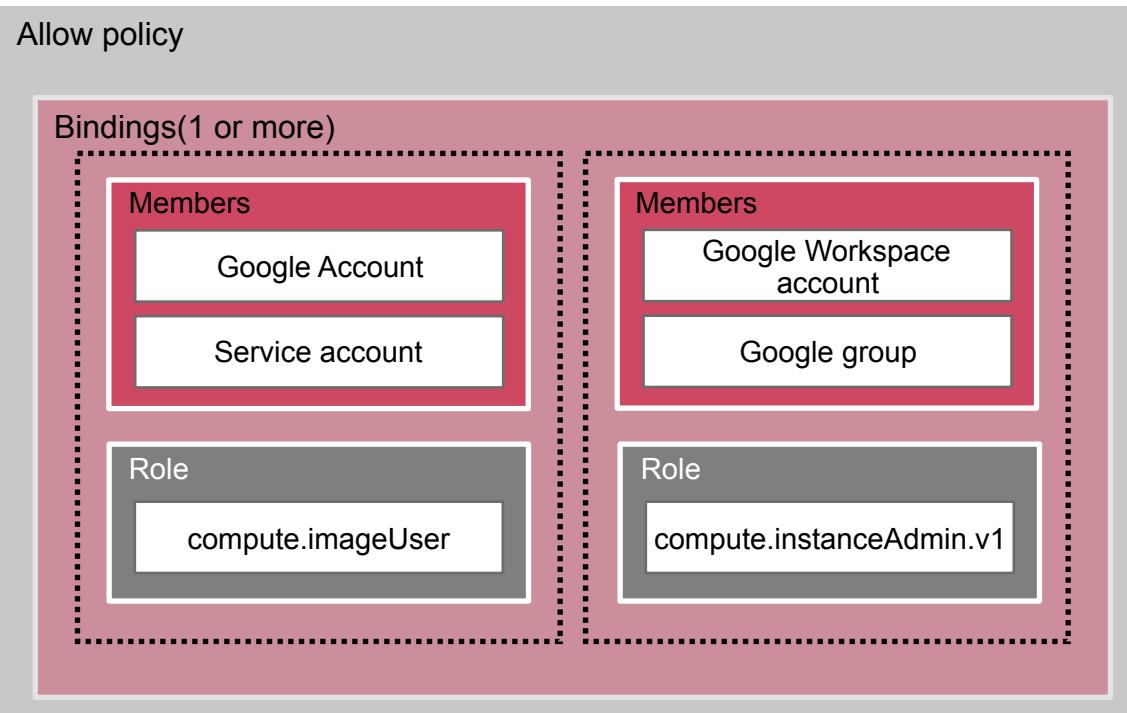
**Deny rules** to set guard-rails on access to Google Cloud resources

- Define deny rules to prevent principals from accessing resources regardless of what roles they are granted
- Deny policies take precedence over allow policies



# Policies can be Applied Anywhere in the Hierarchy

Allow policy

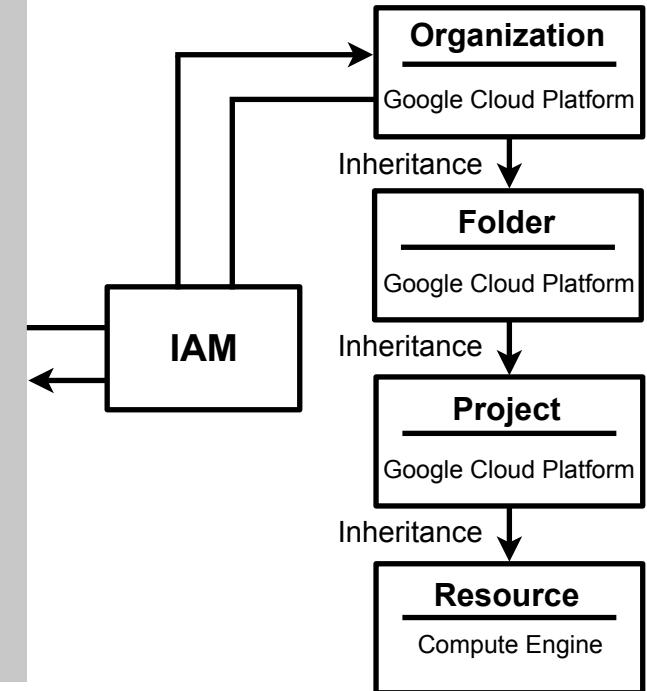
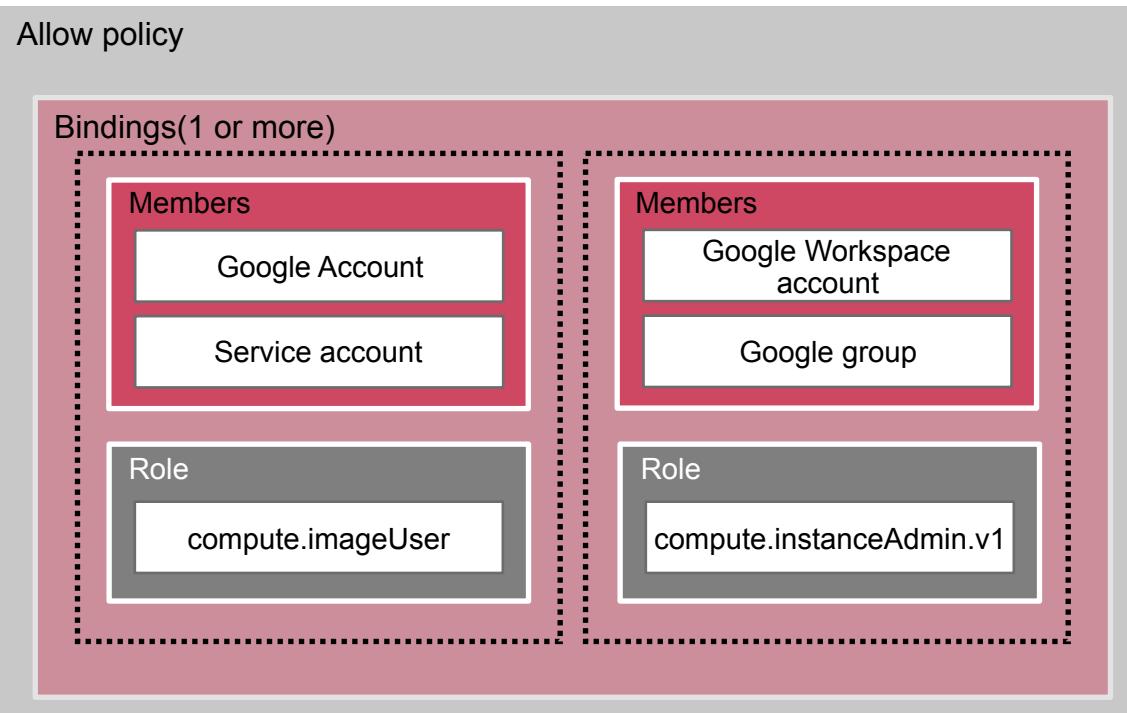


Policies are inherited down the resource hierarchy



# Policies at the Organization Level

Allow policy

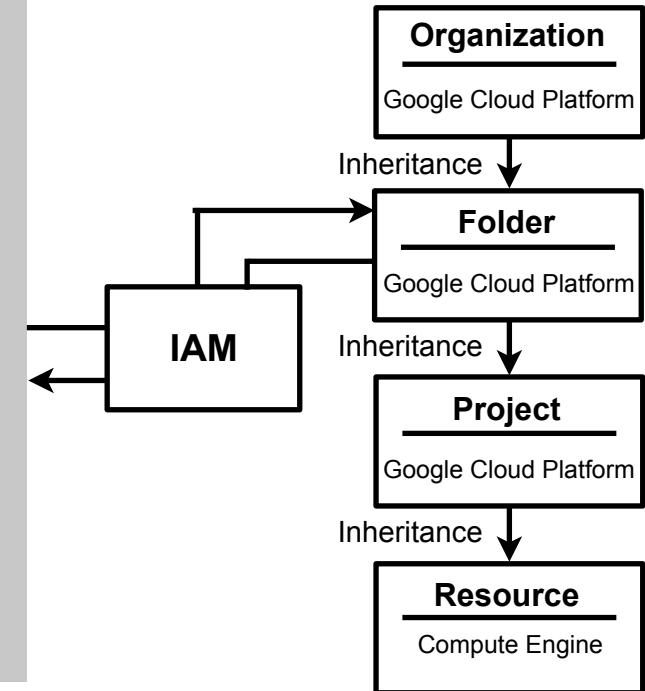
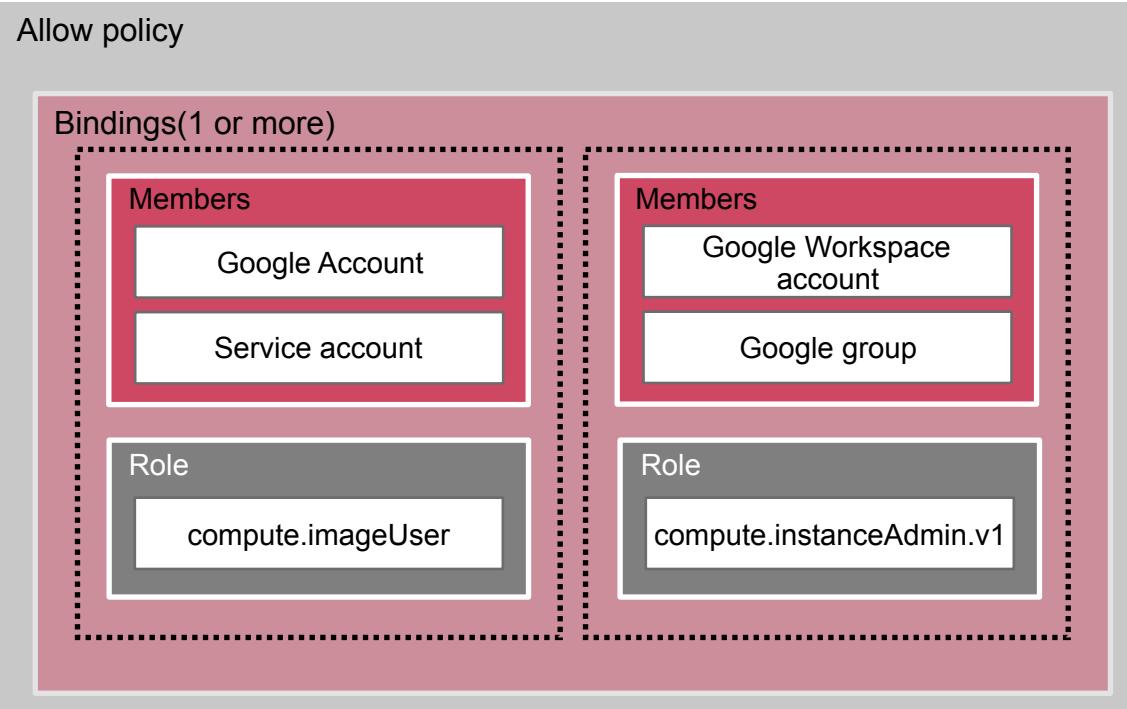


Apply policies at the organization level if policies apply to all departments and all teams in the organization



# Policies at the Folder Level

Allow policy

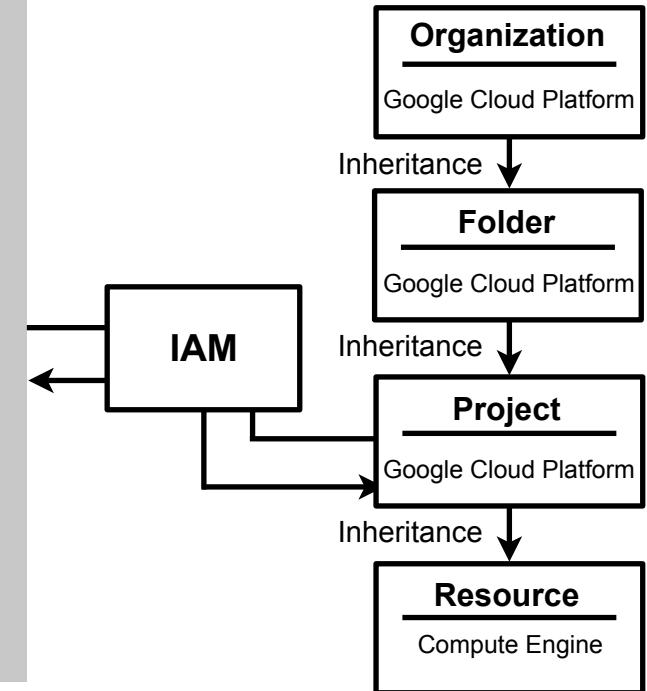
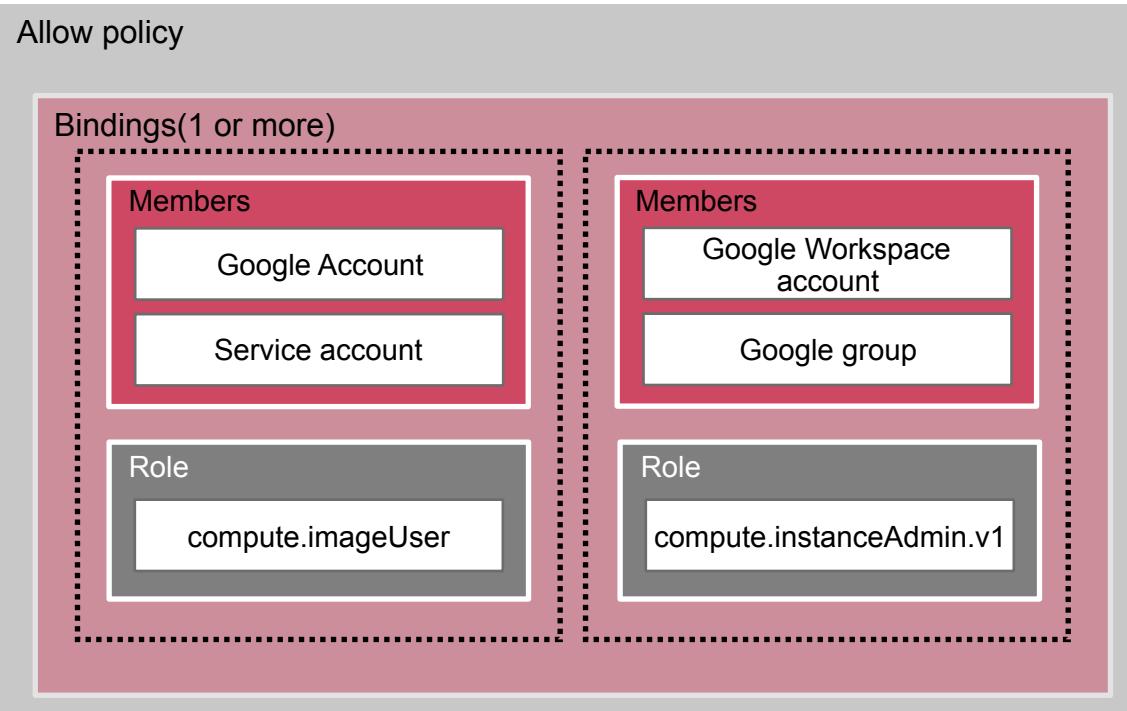


If projects in a department have similar policies apply policies at the folder level and group department projects into folders



# Policies at the Project Level

Allow policy

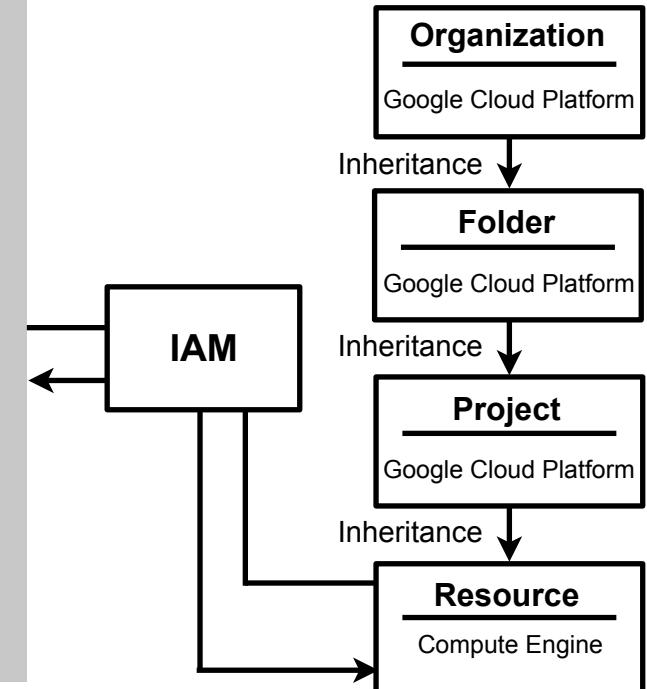
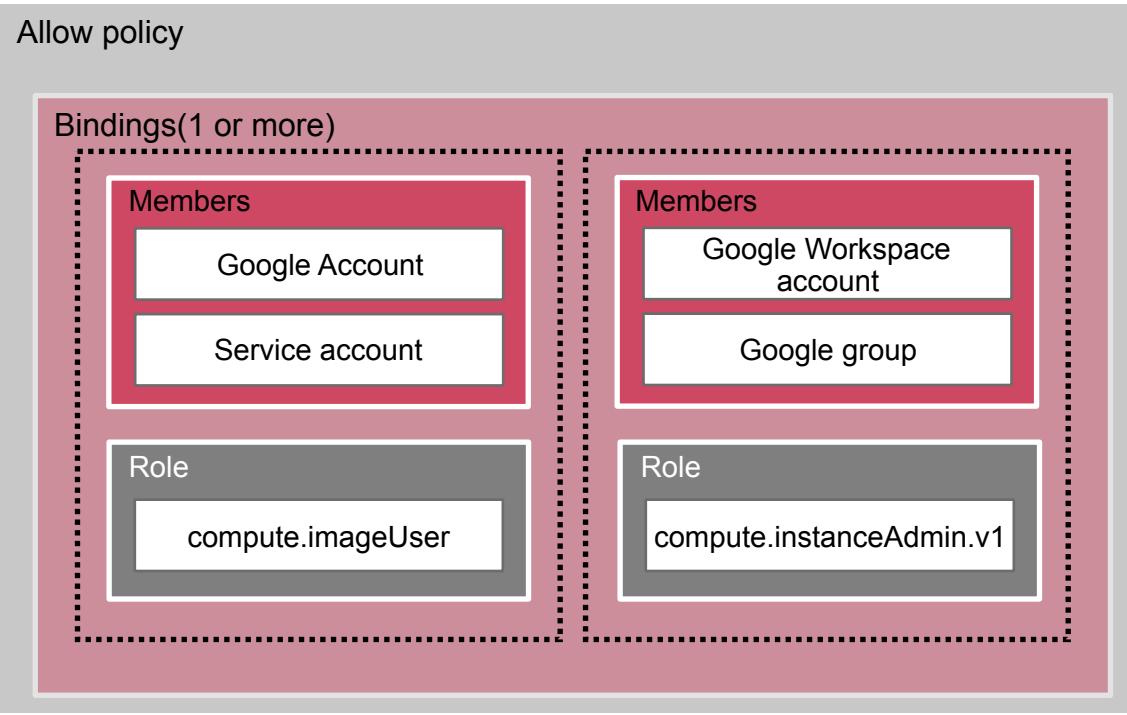


If a certain kind of access is needed only for a specific project  
apply the policy at the project level



# Policies at the Resource Level

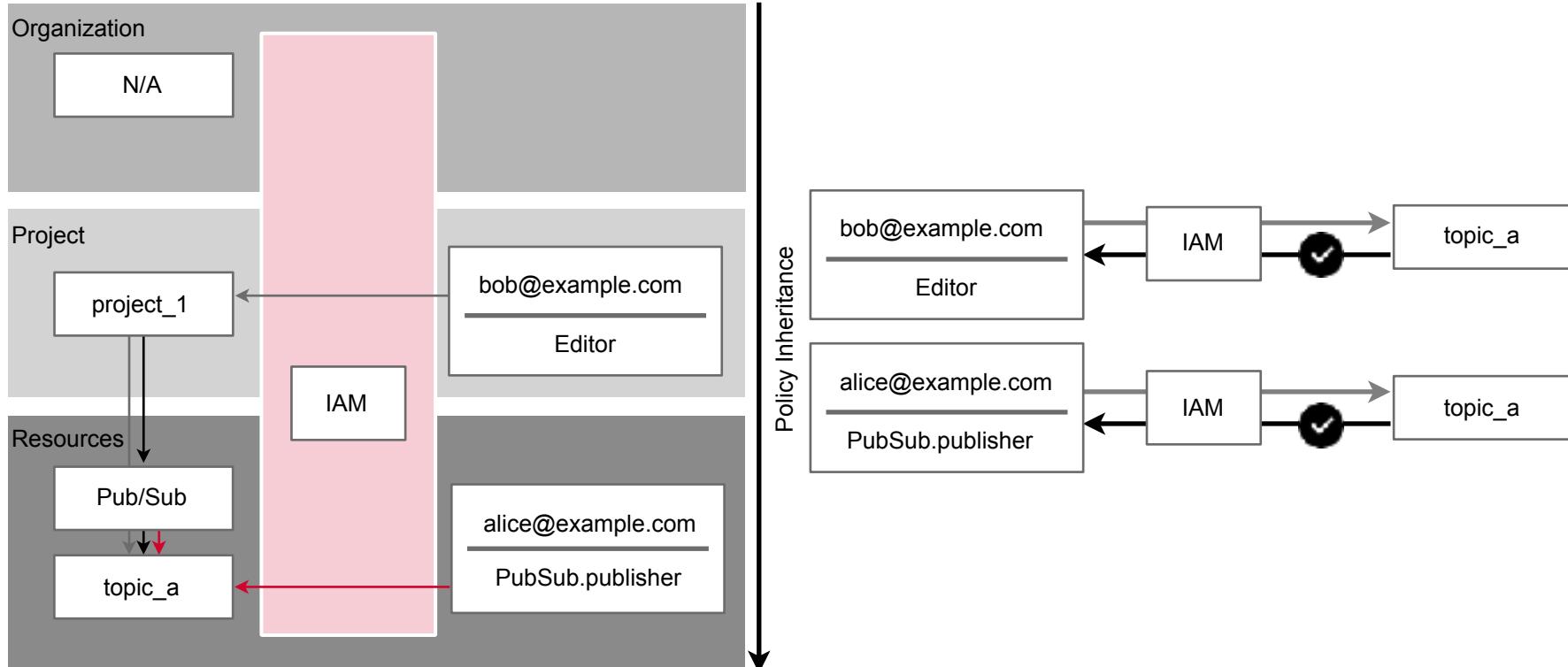
Allow policy



If access is to be granted to a **specific resource and NOT a specific type of resource** - assign policies at the resource level



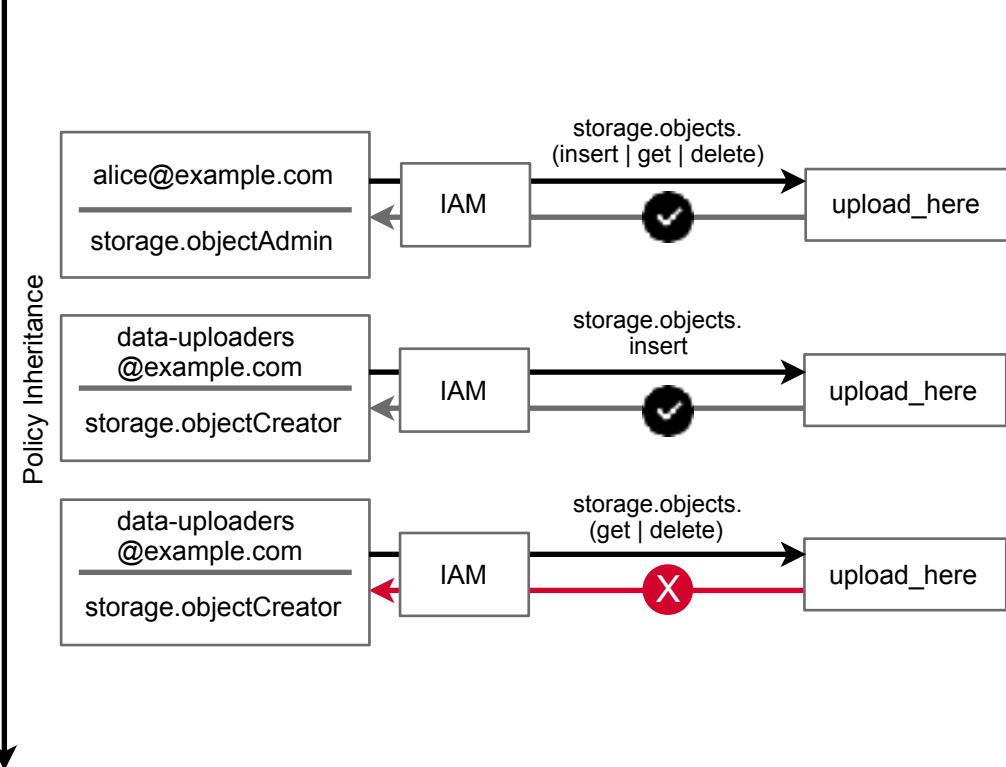
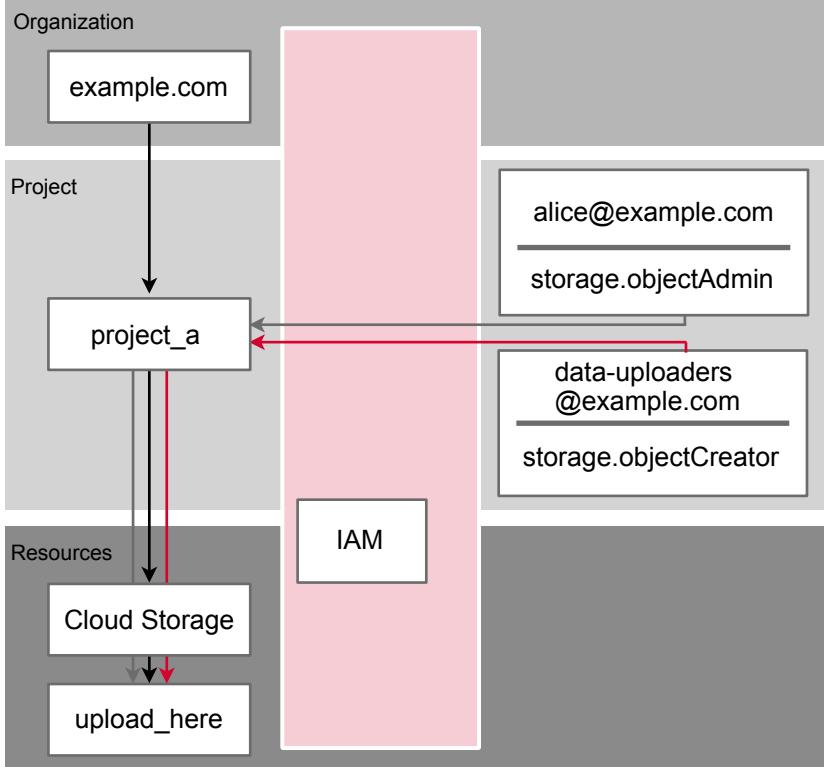
# Allow Policy Inheritance



Bob has full control over topic\_a while Alice can only publish messages to that topic



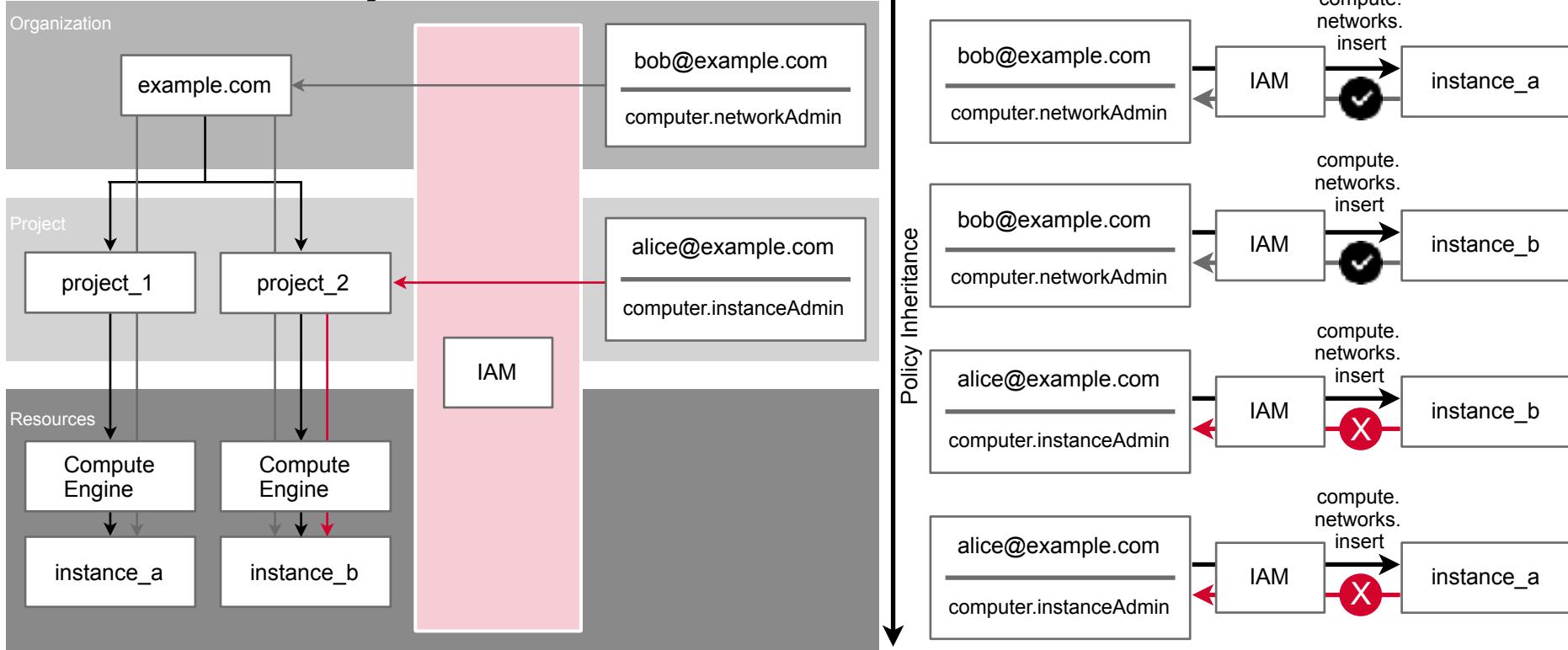
# Allow Policy Inheritance



Alice can insert, get, or delete objects in buckets but members of the data-uploaders group can only insert objects



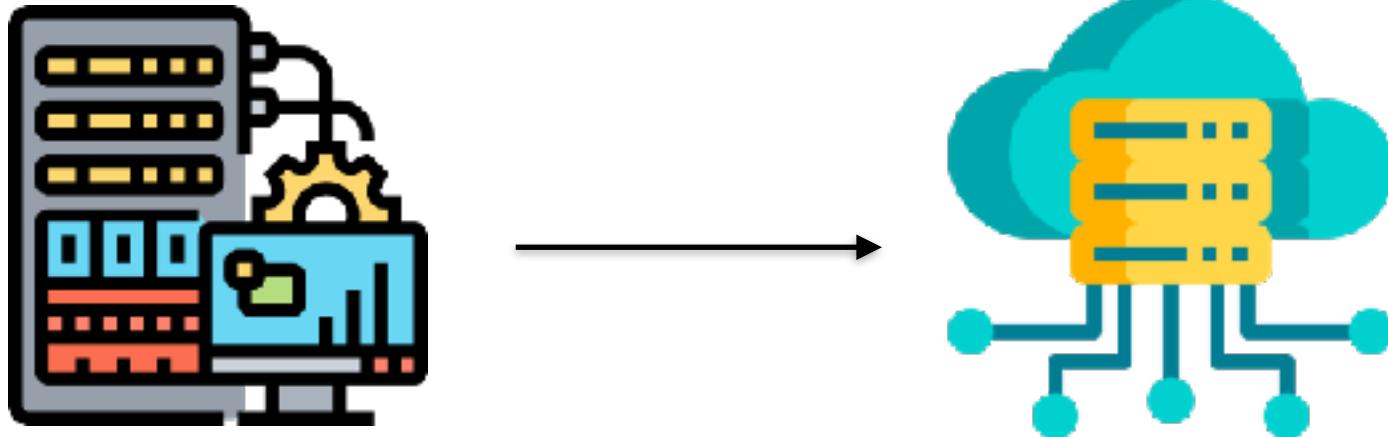
# Allow Policy Inheritance



Bob can administer the networks of all projects in the organization while  
Alice can only manage compute instances in project\_2 not networks



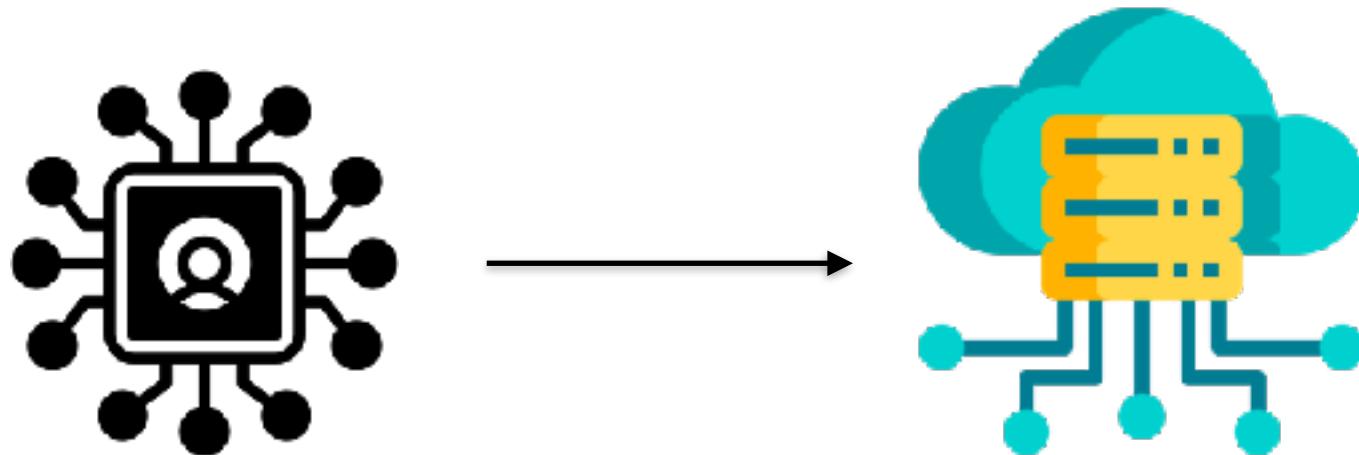
# Applications Accessing Cloud Resources



Your application running on the server needs to access cloud resources as a part of its execution - how do you assign permissions to your application?



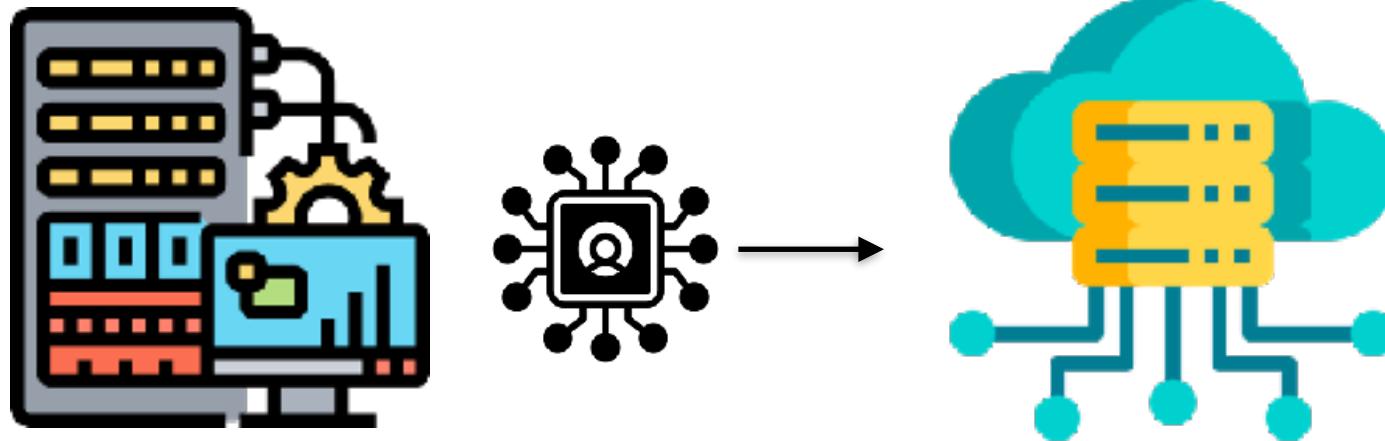
# Service Accounts



Create a service account and give it access  
to the resource on the cloud



# Configure Application to Use the Service Account



The application will run under the identity of the service account and use the roles granted to the service account to access resources on the cloud



**Applications running in one project  
can access services running in  
another project using service  
accounts**



# Accessing Cloud Resources Across Projects



Project A

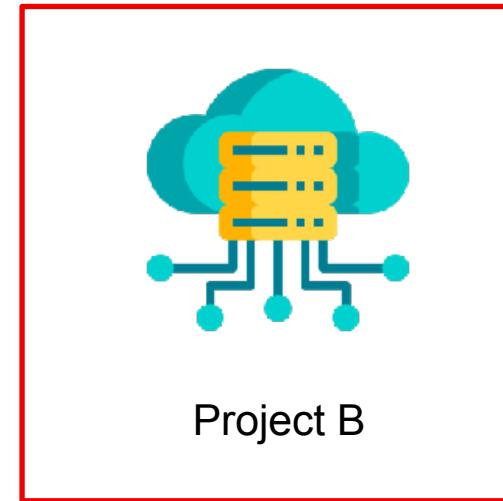
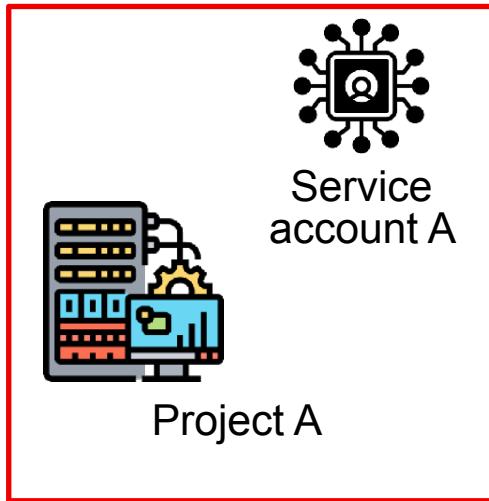


Project B

Application running in project A needs access  
to a database in project B



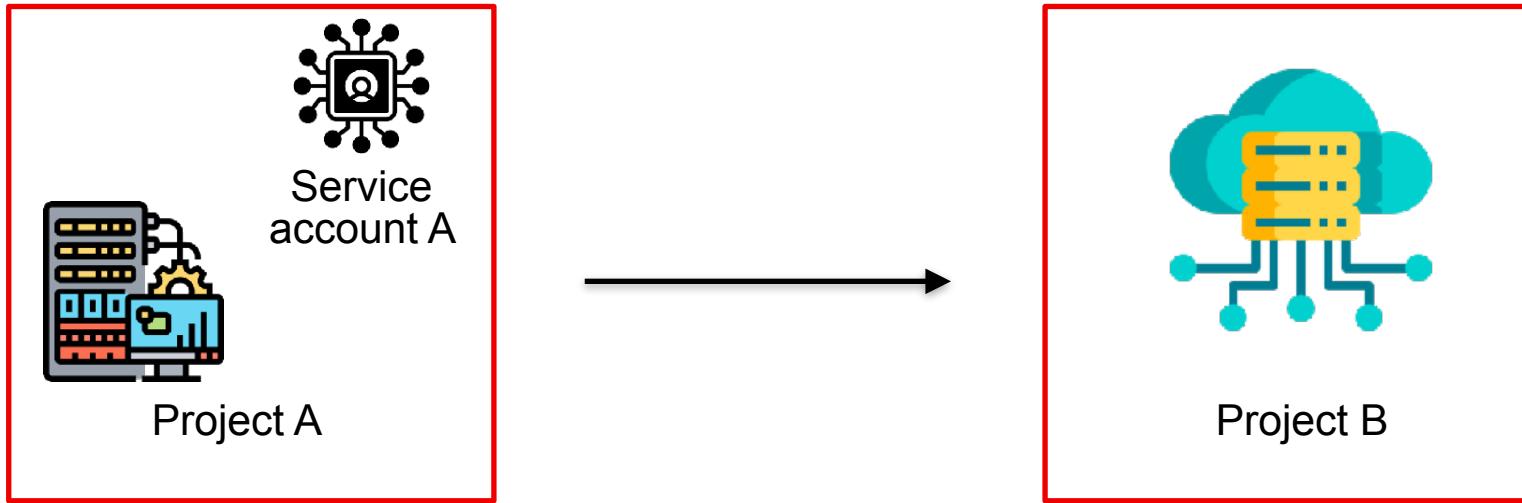
# Create a Service Account in Project A



Project A owners create a service account in Project A



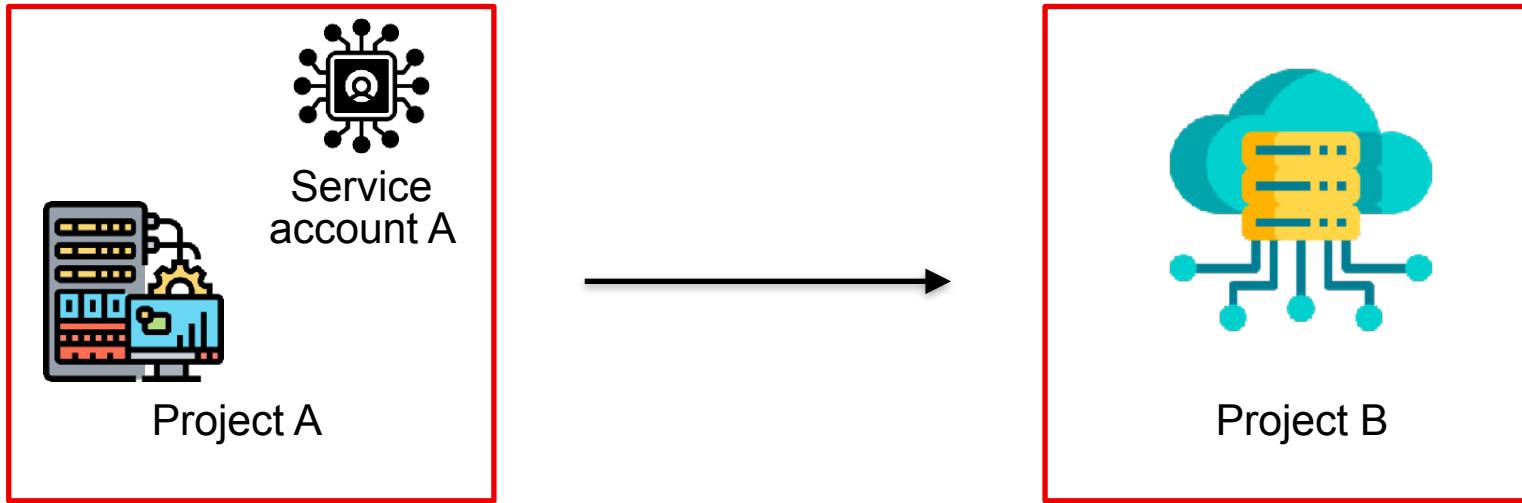
# Configure Service Account to Access Resources



Project B owners give service account A  
access to the database in Project B



# Accessing Cloud Resources Across Projects



Applications running in Project A can now  
access the database in Project B



# Service Account Keys

- Can generate keys which act as credentials to allow applications to authenticate as the service account





# Cloud Identity

- Centralized management for users, groups, roles, devices
- Single Sign On (SSO) support
- Multi-factor Authentication (MFA): Passwords + mobile device verification
- Device management: Tools to manage and secure corporate devices
- Can integrate with external identity providers i.e Microsoft Active Directory
- Synchronize using:
  - Google Cloud Directory Sync (GCDS)
  - Identity Federation





# Google Workspace

- Uses Cloud Identity as an identity management service under the hood
- Includes all features of Cloud Identity
- Provides productivity tools such as Gmail, Google Drive, Google Docs





# Cloud Identity vs. Google Workspace

## Cloud Identity

- Primary focus on IAM for users, devices, and groups
- Services include user management, SSO, MFA and security controls
- Need security management but not productivity tools

## Google Workspace

- Focus on productivity tools (like Gmail, Drive, Docs)
- Includes all features of Cloud Identity + productivity tools
- Both identity management and productivity tools in one package

O'REILLY®

# IAM Best Practices





# Apply the principle of least privilege



# Principle of Least Privilege

Use Case	Role
A user needs to create and manage storage buckets but doesn't require access to the contents of the objects	<b>roles/storage.admin</b> role, which includes permissions like storage.buckets.create, storage.buckets.delete, and storage.buckets.update. User cannot modify contents of buckets
A data analyst needs to view dataset metadata and query results but doesn't need permission to modify the datasets	<b>roles/bigquery.dataViewer</b> role, which includes permissions like bigquery.datasets.get and bigquery.tables.getData. This restricts the user to read-only access, preventing modifications.
A security administrator needs to manage service account permissions but should not have full access to other resources.	<b>roles/iam.serviceAccountAdmin</b> role, which includes iam.serviceAccounts.setIamPolicy, allowing the administrator to manage permissions for service accounts without having access to other resources.



# Best Practices

- Use principle of least privilege to grant the **smallest necessary set of permissions**
- **Grant roles to groups** rather than individual users if possible
  - Easier to manage members in a group rather than individually
- **Grant roles to the smallest scope** needed
  - Need access to publish messages to a topic? Grant the role only on that topic





# Best Practices

- If you need to grant a role to a user or group that **spans projects - grant at the folder level**
- Use audit logs to ensure compliance
  - Allow you to trace when a policy has been created or modified
- Use IAM on the Google Cloud Console to view the latest roles granted to each user



O'REILLY®

# Organization Policy Service





# Organization Policy Service

Gives you centralized and programmatic control over your organization's cloud resources.

The organization administrator can configure **constraints** across the entire resource hierarchy.

Can be applied at organization, folder, or project level



# Examples of Organization Policies

- **Restrict resource locations**

- Helps meet compliance and storage requirements by ensuring resources created only in permitted regions

- **Control service account key creation**

- Restricts service account key creation ensuring other more secure methods of authentication are used

- **Enforce Cloud Storage bucket uniform access:**

- Disables buckets with fine-grained permissions for individual objects



# IAM

---

You are setting up service accounts for a distributed application across multiple Google Cloud projects. Virtual machines (VMs) in the data-analytics project require access to Cloud Storage buckets in the shared-resources project. To adhere to Google-recommended best practices, how should you configure access for the service account in the data-analytics project?

- A. Assign the "project editor" role to the data-analytics project and "roles/storageObjectAdmin" to shared-resources.
- B. Grant the "roles/storage.objectViewer" role to shared-resources and ensure that the apps running on VMs in data-analytics have the right permissions
- C. Assign the "project editor" role to both the data-analytics and shared-resources projects.
- D. Assign the "project editor" role to shared-resources and grant the "roles/storage.objectViewer" role to data-analytics.



# IAM

---

You are setting up service accounts for a distributed application across multiple Google Cloud projects. Virtual machines (VMs) in the data-analytics project require access to Cloud Storage buckets in the shared-resources project. To adhere to Google-recommended best practices, how should you configure access for the service account in the data-analytics project?

- A. Assign the "project editor" role to the data-analytics project and "roles/storageObjectAdmin" to shared-resources.
- B. Grant the "roles/storage.objectViewer" role to shared-resources and ensure that the apps running on VMs in data-analytics have the right permissions**
- C. Assign the "project editor" role to both the data-analytics and shared-resources projects.
- D. Assign the "project editor" role to shared-resources and grant the "roles/storage.objectViewer" role to data-analytics.



---

# IAM

You are deploying a new virtual machine (VM) on Google Cloud to run a specific batch processing job. The VM needs to access a Cloud Storage bucket and a Cloud SQL instance, but you want to ensure it only has the necessary permissions for these tasks following Google Cloud's best practices. What should you do?

- A. Use the default Compute Engine service account and modify its permissions
- B. Assign the "project owner" role to the VM instance for full access.
- C. Create a new service account with the right permissions and use this service account while creating the VM that runs the job.
- D. Manually configure the VM to use your personal credentials for accessing resources.



# IAM

---

You are deploying a new virtual machine (VM) on Google Cloud to run a specific batch processing job. The VM needs to access a Cloud Storage bucket and a Cloud SQL instance, but you want to ensure it only has the necessary permissions for these tasks following Google Cloud's best practices. What should you do?

- A. Use the default Compute Engine service account and modify its permissions
- B. Assign the "project owner" role to the VM instance for full access.
- C. Create a new service account with the right permissions and use this service account while creating the VM that runs the job.**
- D. Manually configure the VM to use your personal credentials for accessing resources.



# IAM

---

A security auditor has been brought in and wishes to review who has the ability to edit resources across your entire project. What is the best way to do this?

- A. Check each compute engine instance to see whether SSH keys have been stored
- B. Enable the Recommender API and review its security recommendations regarding IAM roles and permissions.
- C. Enable and review Audit Logs on the IAM & Admin page for all resources and export these to BigQuery for the auditor
- D. Go to the Cloud Console IAM & Admin page or use the gcloud tool to see who has the “Project Editor” role assigned



# IAM

---

A security auditor has been brought in and wishes to review who has the ability to edit resources across your entire project. What is the best way to do this?

- A. Check each compute engine instance to see whether SSH keys have been stored
- B. Enable the Recommender API and review its security recommendations regarding IAM roles and permissions.
- C. Enable and review Audit Logs on the IAM & Admin page for all resources and export these to BigQuery for the auditor
- D. Go to the Cloud Console IAM & Admin page or use the gcloud tool to see who has the “Project Editor” role assigned**



# IAM

---

You are overseeing multiple projects within the marketing department on Google Cloud. To enhance security and maintain control over service account usage, you need to ensure that no one in the marketing projects can create new service accounts without proper oversight. What is the best way to enforce this?

- A. Manually remove service account creation permissions from all users in each marketing project.
- B. Configure an organization policy on the folder for all marketing projects to disable service account creation.
- C. Set up Identity-Aware Proxy to block service account access for all marketing projects.
- D. Assign a custom role with limited permissions for managing service accounts across all marketing projects.



# IAM

---

You are overseeing multiple projects within the marketing department on Google Cloud. To enhance security and maintain control over service account usage, you need to ensure that no one in the marketing projects can create new service accounts without proper oversight. What is the best way to enforce this?

- A. Manually remove service account creation permissions from all users in each marketing project.
- B. Configure an organization policy on the folder for all marketing projects to disable service account creation.**
- C. Set up Identity-Aware Proxy to block service account access for all marketing projects.
- D. Assign a custom role with limited permissions for managing service accounts across all marketing projects.



# Billing Accounts





# Cloud Billing

Collection of tools that help you track and understand your Google Cloud spending, pay your bill, and optimize your costs



# Billing Account

Cloud resource that defines who pays for Google Cloud resources. A single invoice is generated per cloud billing account

Billing accounts are linked to Google Payments Profile



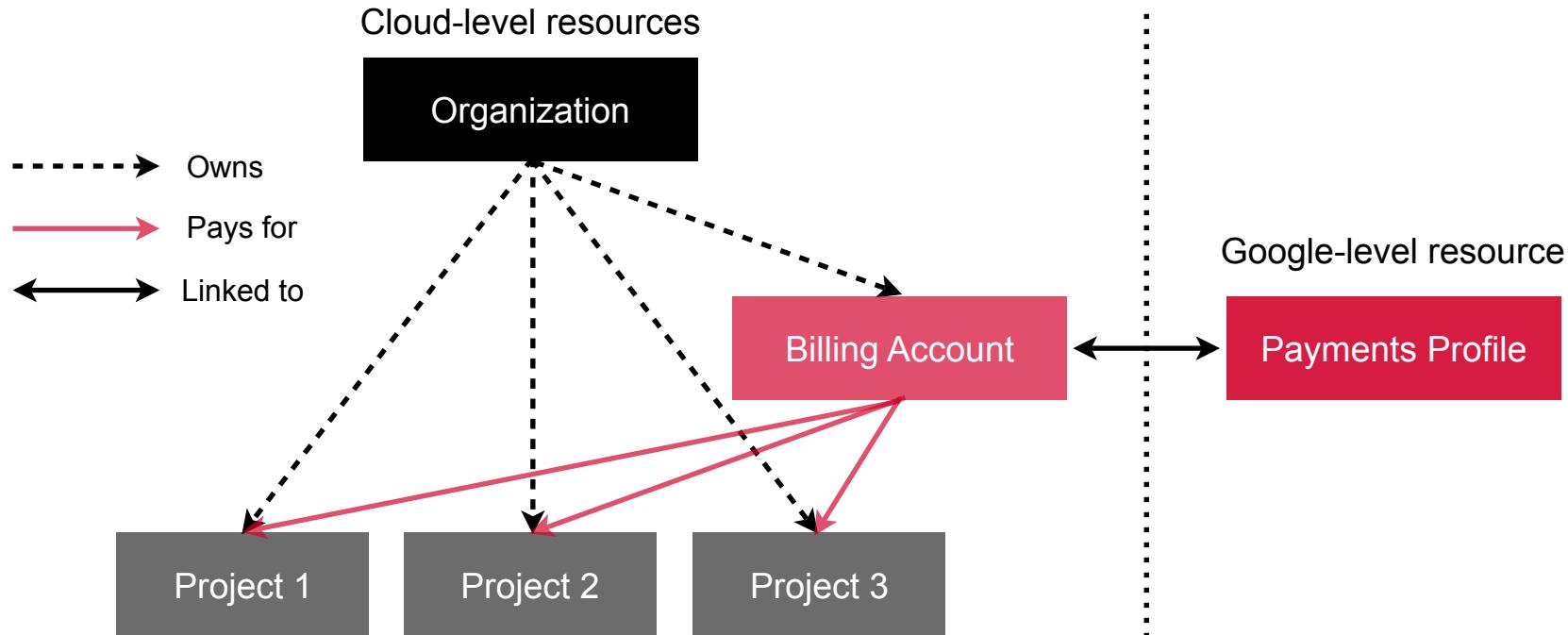
# Payments Profile

Google-level resource that holds a payment instrument that can be charged. Processes payments for ALL Google services

- Cloud services
- Ads
- Maps



# Billing Accounts and Payment Profile



If you want a single invoice for multiple projects across your organization link all the projects to a single billing account



# Cloud Billing Budgets

Budgets help you track your actual Google Cloud costs against your planned costs.

Allow you to configure threshold rules to trigger email notifications



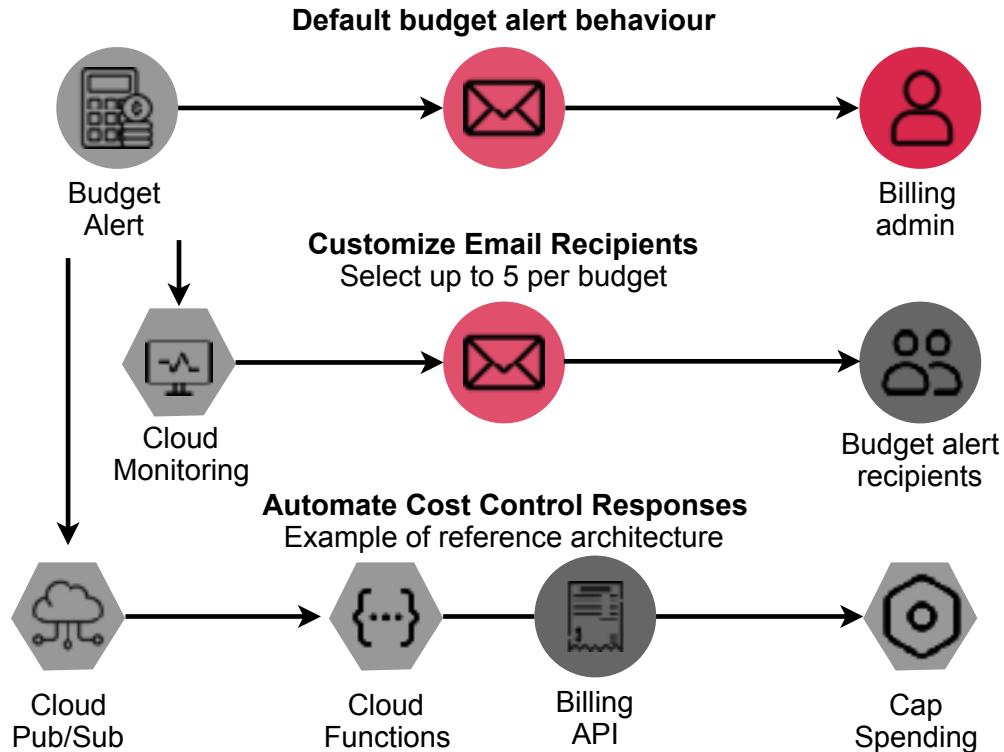
# Cloud Billing Budgets

- Configure budgets for specific time ranges e.g. monthly, quarterly, yearly
- Define the scope of budgets:
  - Entire billing account
  - Per organization, folder, project
  - One or more services e.g. BigQuery, BigTable
  - Resources with a specific label



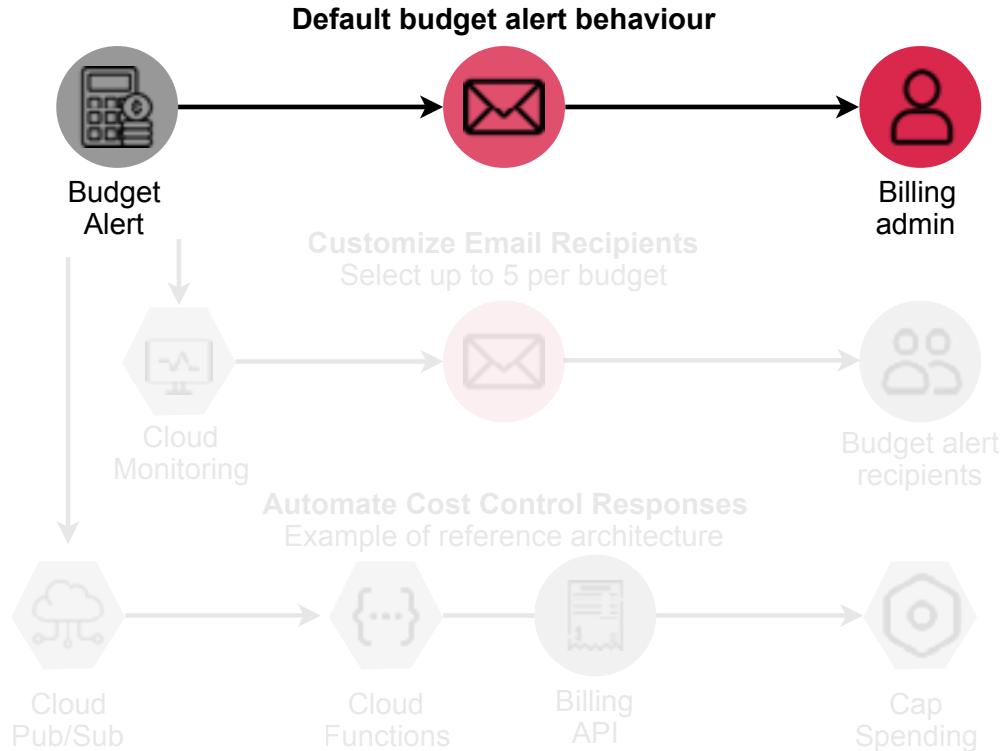


# Budget Notifications



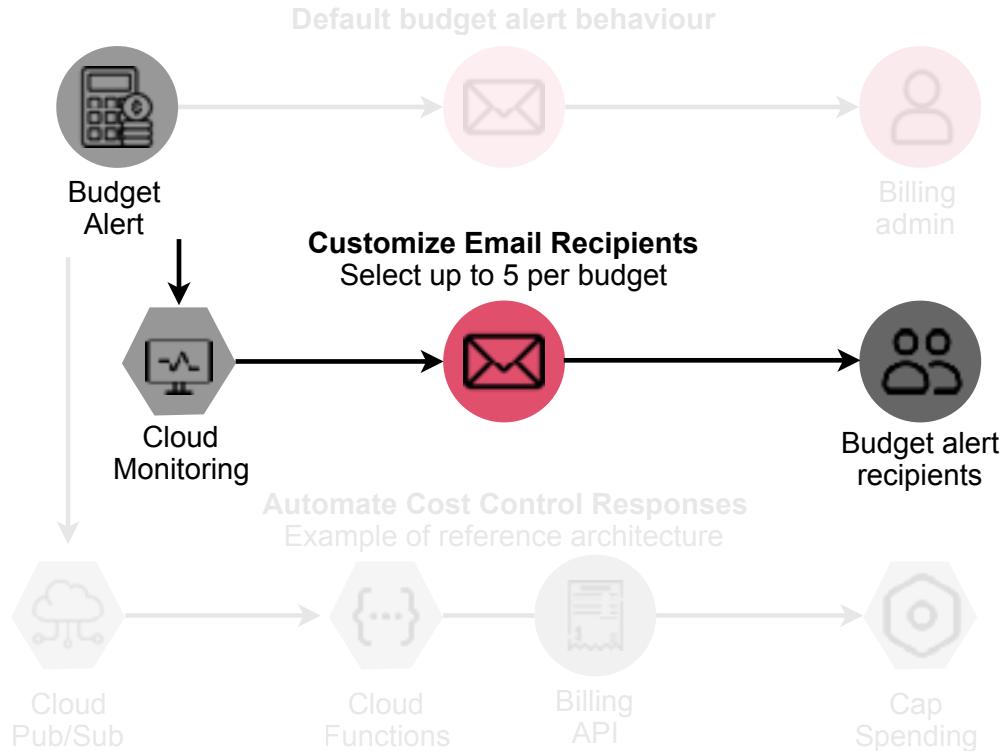


# Notify Billing Admin and Billing Users



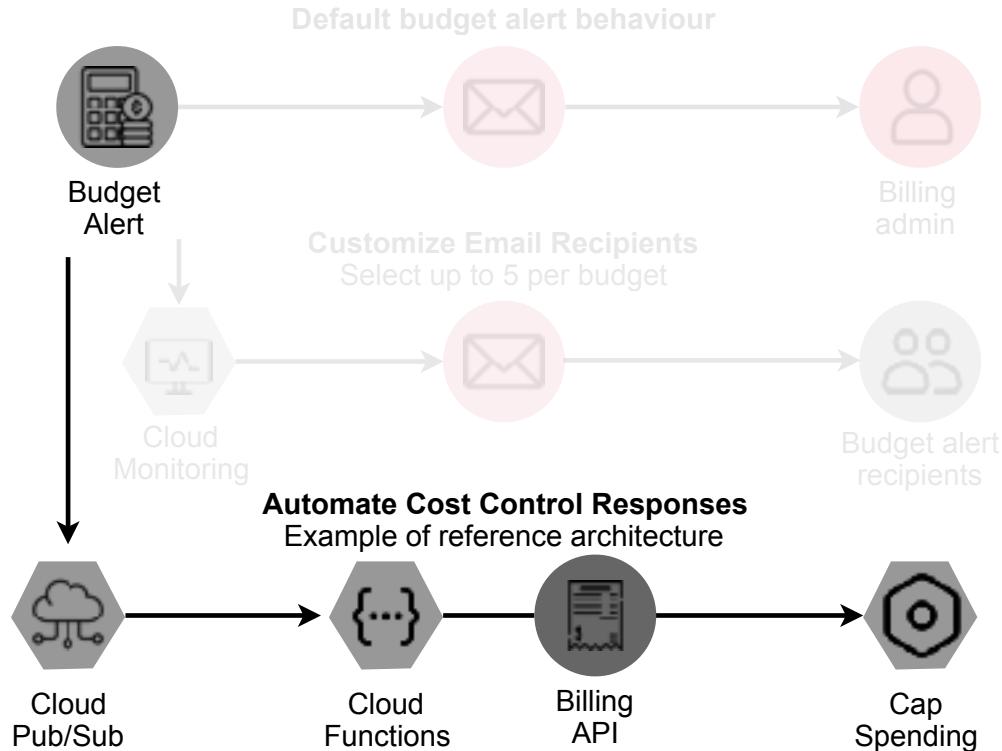


# Cloud Monitoring to Notify Others in Your Org





# Pub/Sub for Any Programmatic Notifications





# Export Billing to BigQuery

- Billing details can be exported to BigQuery for detailed analysis
- Allows **custom queries** that may not be supported by Cloud Billing on the Cloud Console
- Can break down cost based on labels
- Labels used to
  - Resources in different environments
  - Resources used for specific products
- Can use **Looker Studio to build a dashboard** to visualize billing information





# Important Cloud Billing Roles

Role	Purpose
Billing Account Creator (roles/billing.creator)	Create new self-serve (online) billing accounts.
Billing Account Administrator (roles/billing.admin)	Manage billing accounts (but not create them).
Billing Account Costs Manager (roles/billing.costsManager)	Manage budgets and view and export cost information of billing accounts (but not pricing information).
Billing Account Viewer (roles/billing.viewer)	View billing account cost information and transactions.
Billing Account User (roles/billing.user)	Link projects to billing accounts.
Project Billing Manager (roles/billing.projectManager)	Link/unlink the project to/from a billing account.

# Billing

Your company has been using Google Cloud for various teams, with each project owner managing their own billing accounts separately. As a result, billing has been ad-hoc, and the finance department struggles to consolidate cloud expenses across different projects. The company leadership now wants to streamline billing by having a single invoice for all projects across the organization. What should you do to achieve this?

- A. Create a single billing account for the organization and ask all project owners to link their projects to this billing account.
- B. Set up billing export to BigQuery and create a custom report to track all expenses.
- C. Instruct each project owner to manually export their billing data and send monthly reports to finance.
- D. Enable cost management and budgeting for each project individually to monitor costs separately.



# Billing

Your company has been using Google Cloud for various teams, with each project owner managing their own billing accounts separately. As a result, billing has been ad-hoc, and the finance department struggles to consolidate cloud expenses across different projects. The company leadership now wants to streamline billing by having a single invoice for all projects across the organization. What should you do to achieve this?

- A. Create a single billing account for the organization and ask all project owners to link their projects to this billing account.**
- B. Set up billing export to BigQuery and create a custom report to track all expenses.
- C. Instruct each project owner to manually export their billing data and send monthly reports to finance.
- D. Enable cost management and budgeting for each project individually to monitor costs separately.

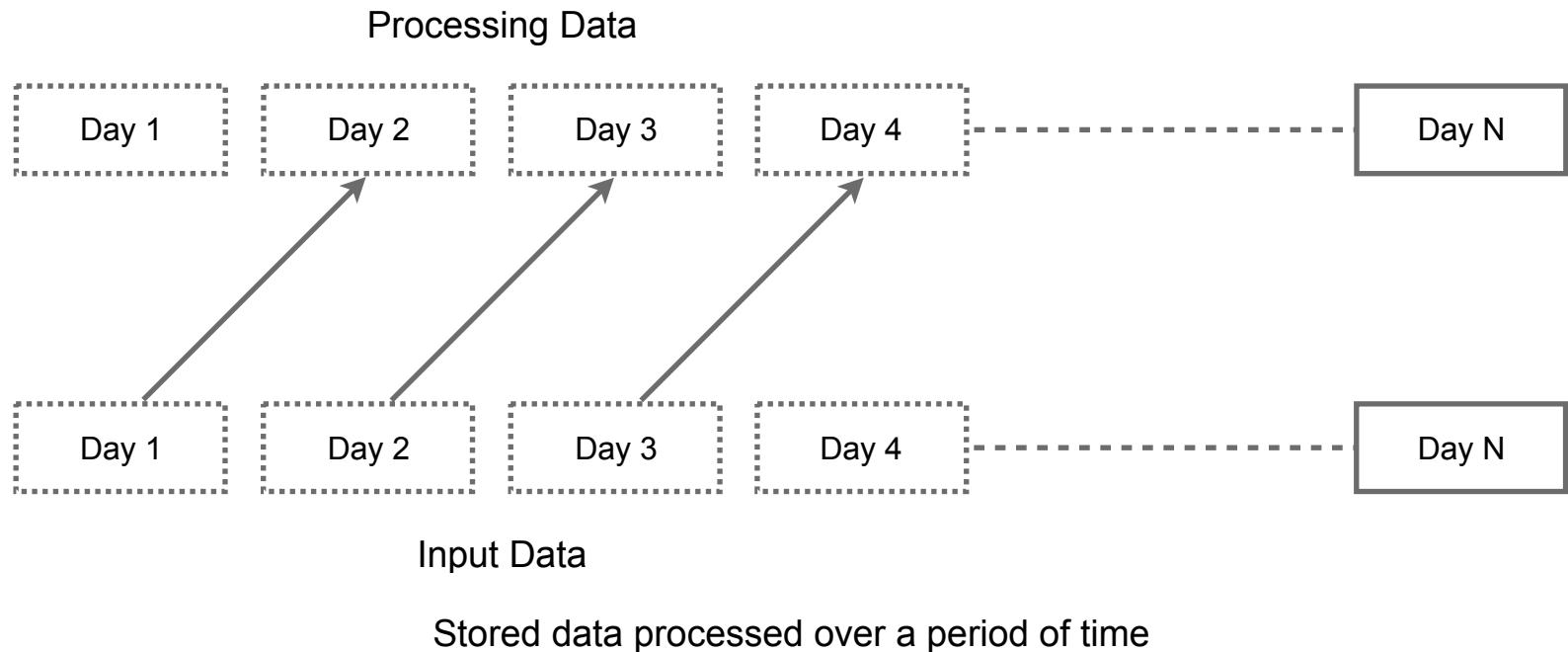


# Batch Processing vs. Stream Processing



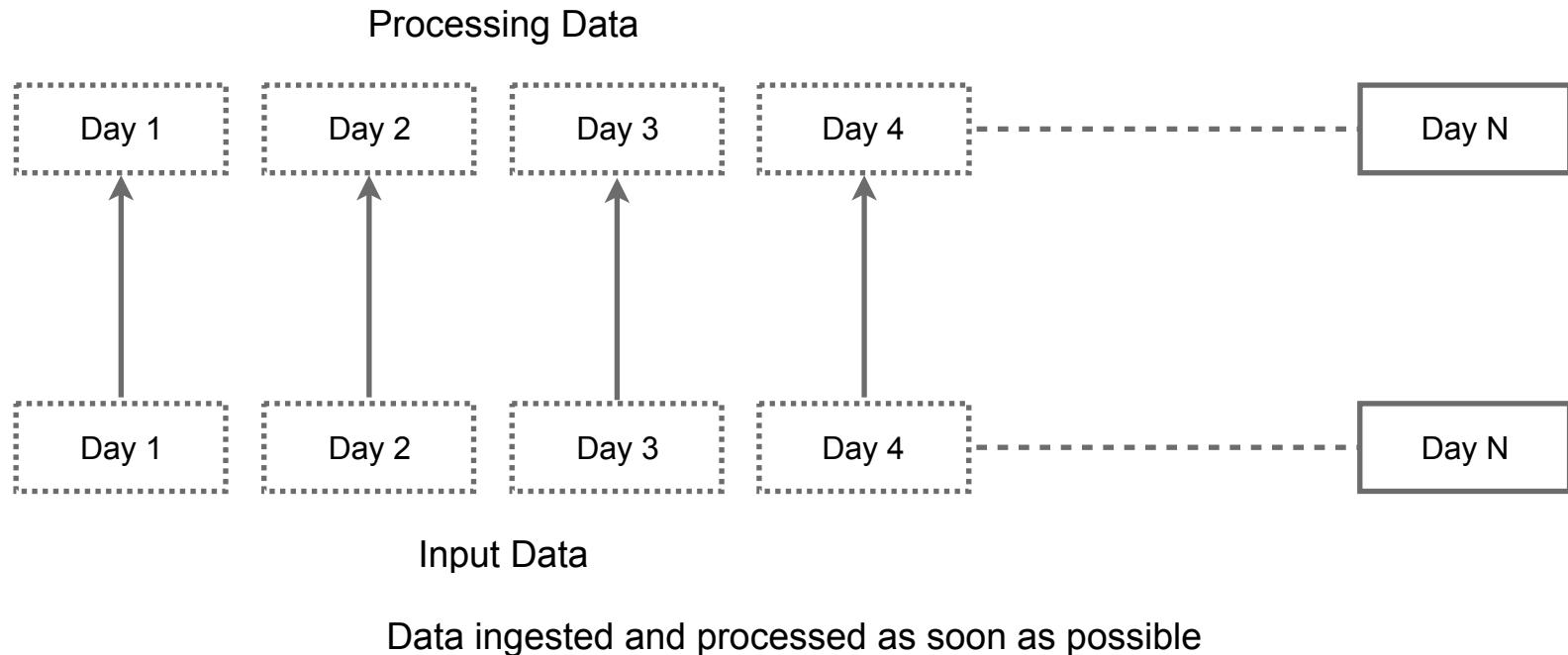


# Batch Processing





# Stream Processing





# Batch and Stream Processing on Google Cloud

Cloud Dataproc

Cloud Dataflow



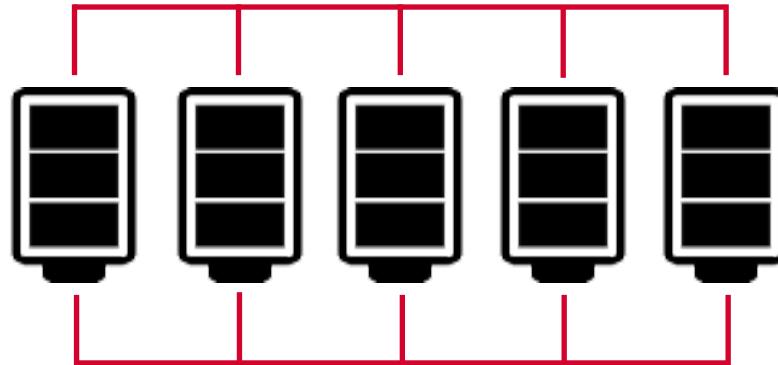
# Batch and Stream Processing on Google Cloud

Cloud Dataproc

Cloud Dataflow



# Hadoop Framework for Distributed Computing



**Runs big data processing applications  
on a cluster of machines**



# Hadoop - Single Coordinating Software

HDFS

MapReduce

YARN

**Storage**

**Compute**

**Co-ordination**



# Hadoop to Dataproc

Google Cloud  
Storage (preferable\*)

Google Compute  
Engine VM instances

Dataproc service -  
Dataproc + YARN

**Storage**

**Compute**

**Co-ordination**



# **Store data on Cloud Storage buckets**

# **Use clusters only for compute exactly when you need them**



# Hadoop vs. Dataproc

## Traditional On-premise Hadoop

- Clusters up and running at all times
- HDFS runs on cluster nodes
- Data for jobs stored in HDFS reads
- Cluster encapsulates state

## Managed Hadoop with Dataproc

- Clusters created and used dynamically
- HDFS runs on persistent disks of VMs
- Data ideally in GCS buckets
- Cluster ideally stateless



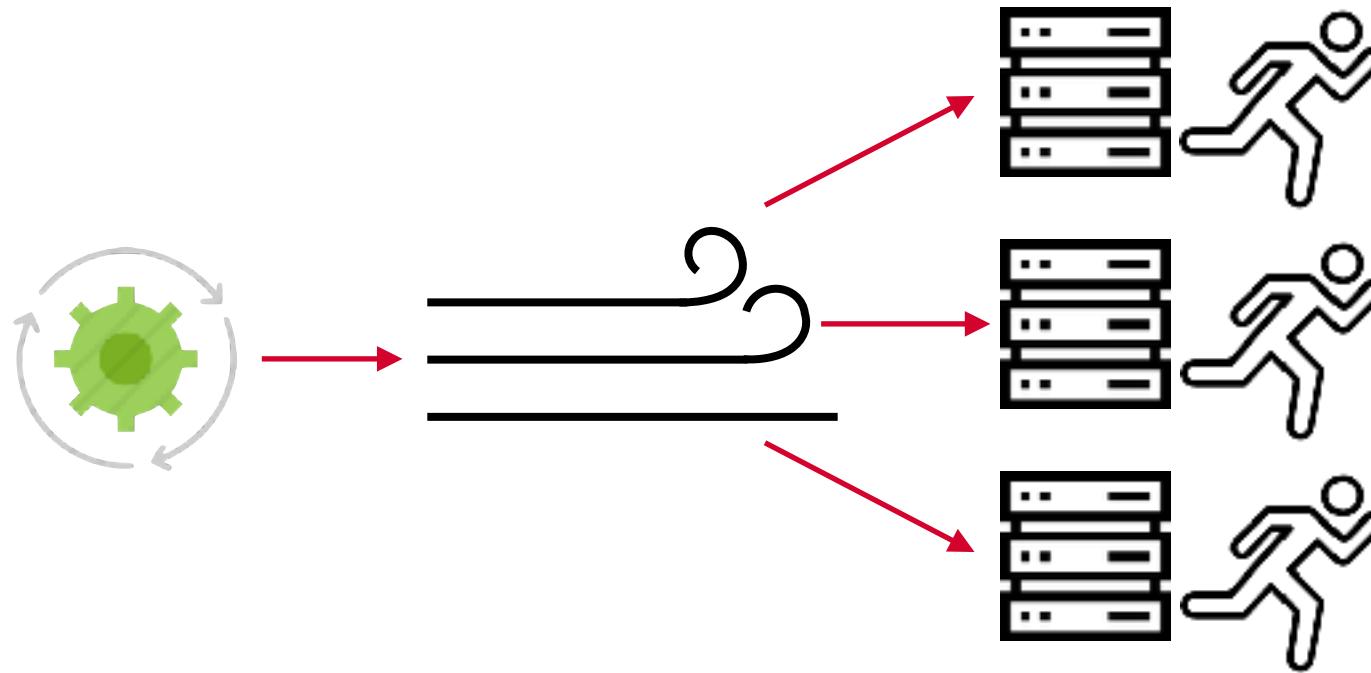
# Dataproc Serverless

- Run Spark workloads without provisioning and managing a Dataproc cluster
- Supports batch workloads as well as interactive workloads
- No infrastructure control or management
- Planned GPU support in the future
- Fast startup times





# Dataproc Serverless





# Batch and Stream Processing on Google Cloud

Cloud Dataproc

Cloud Dataflow



**Cloud Dataflow is a serverless  
approach for data transformation  
of batch as well as streaming data**



# Using Dataflow



Write code for  
pipeline



Submit job for  
execution



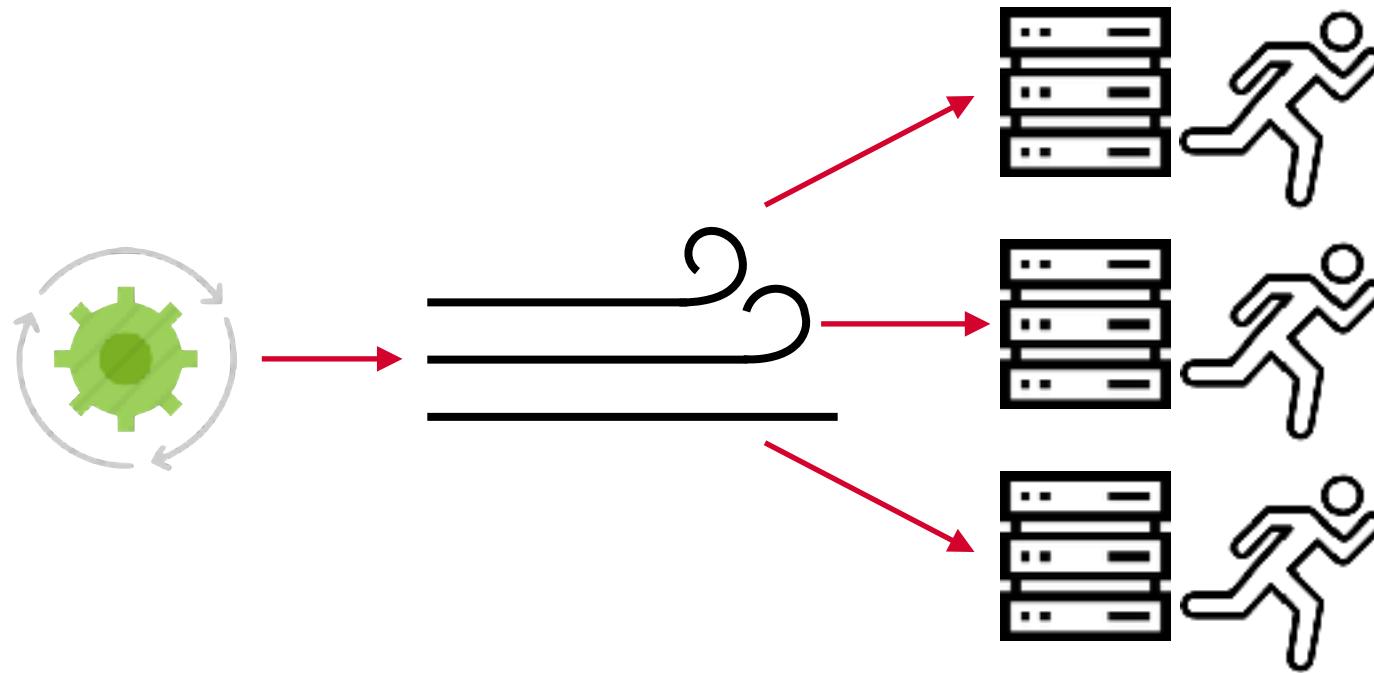
Dataflow assigns  
workers to execute



Pipeline parallelized  
and executed



# Execution Using Workers





# Dataproc vs. Dataflow

## Dataproc

- Requires cluster provisioning
- Manual (but managed) scaling
- Hadoop ecosystem
- Inherently batch-oriented (can work with streaming in Spark)
- Configure infrastructure
- Separation of compute and storage

## Dataflow

- Serverless, no clusters provisioned
- Autoscaling
- Apache Beam API
- Integrates batch and streaming (with Pub/Sub and windowing)
- No access to underlying infrastructure
- Separation of compute and storage



**Choose Dataproc Serverless if you  
want to work with Spark – choose  
Dataflow for Apache Beam APIs and  
other real-time features**

# Pub/Sub





# Cloud Pub/Sub

Simple, global, reliable, asynchronous, real-time messaging and stream ingestion service on the GCP.



# Pub/Sub

- Many-to-many asynchronous messaging
- Decouples senders and receivers
- Reliable, scalable, secure
- At-least-once delivery
- Exactly-once processing (in Dataflow)





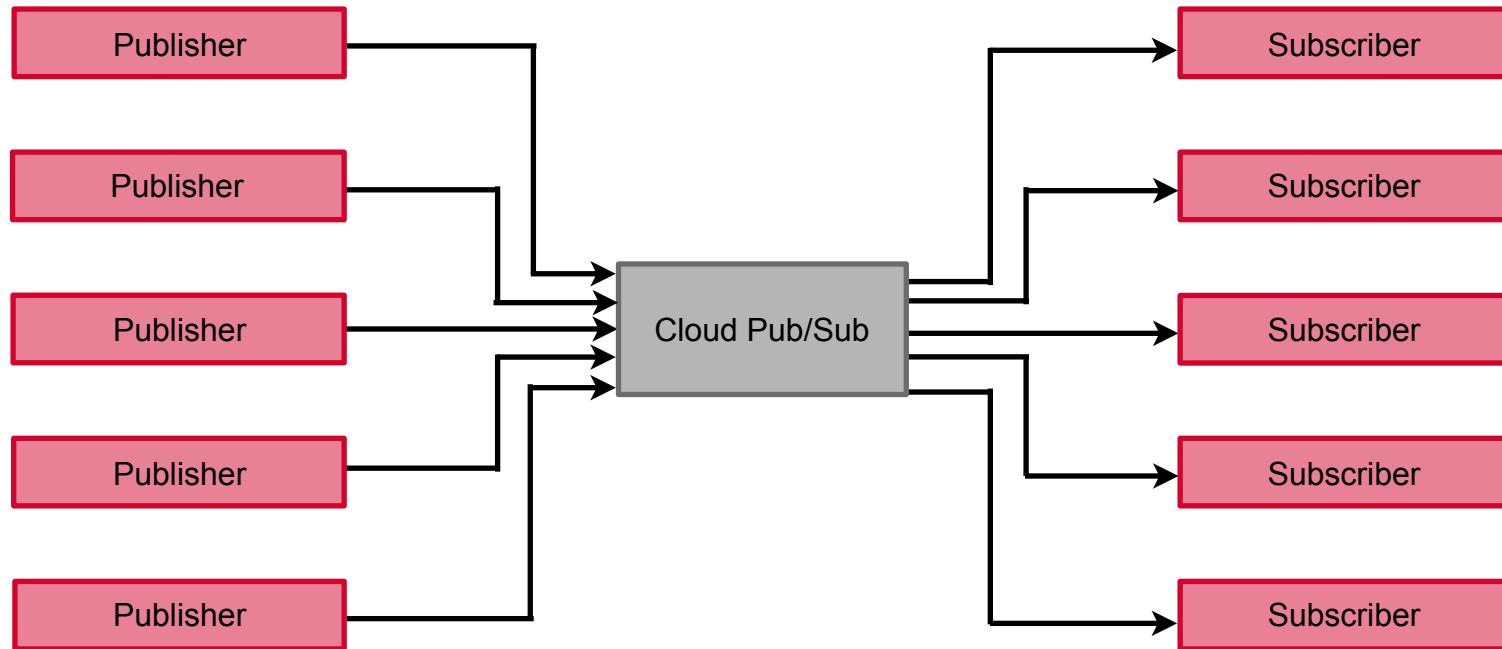
# Pub/Sub

- Serverless
- Autoscaling along all axes
  - Senders
  - Receivers
  - Number and size of messages
- Replicated
- Load-balanced internally





# Messaging Middleware





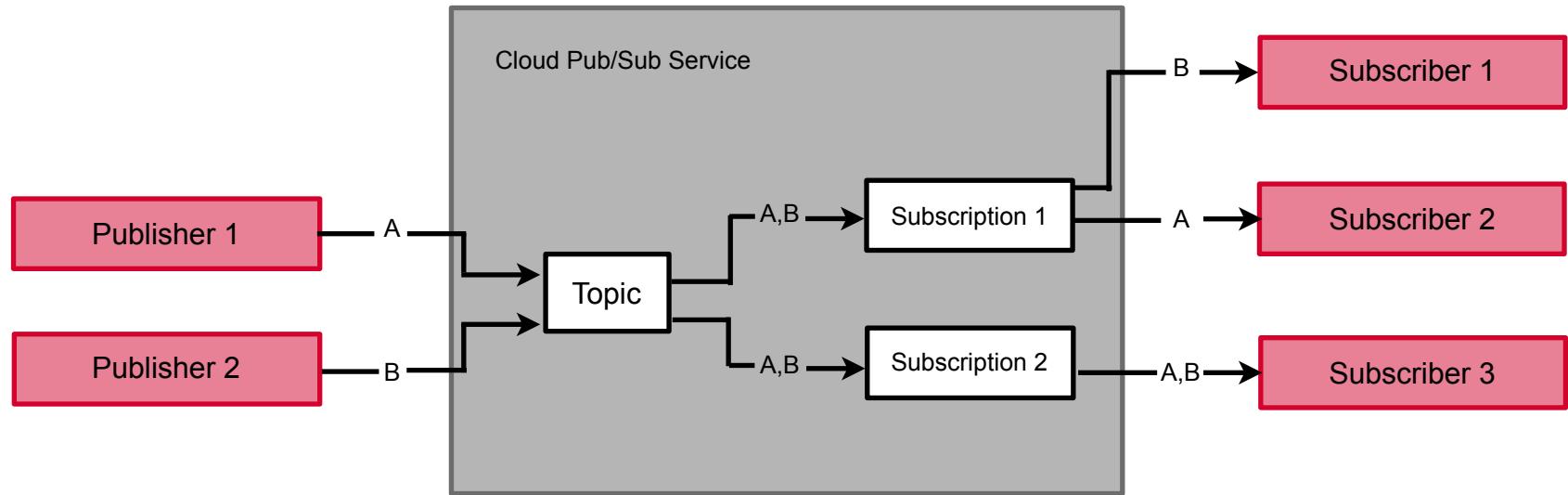
# Pub/Sub

- **Topic**: A named resource to which messages are sent
- **Publisher**: An entity which sends messages to a topic
- **Subscription**: A stream of messages to a topic i.e. a message queue to be delivered to some receiver
- **Message**: A combination of data and attributes
- **Message attribute**: A key-value pair associated with a message





# Publishers, Topics, Subscriptions



# Multiple Services

You are preparing to migrate your on-premises and cloud-based data to Google Cloud. Your data includes:

- 200 TB of media files currently stored in a network-attached storage (NAS) system.
- A data warehouse hosted on Amazon Redshift containing analytics data.
- 20 GB of image files currently stored in an Amazon S3 bucket.

You need to transfer the media files to a Cloud Storage bucket, migrate the data warehouse to BigQuery, and move the image files to a separate Cloud Storage bucket. You want to use Google-recommended tools and avoid custom coding for the migration. What should you do?

- A. Use Transfer Appliance to move the media files, BigQuery Data Transfer Service for Redshift migration, and Storage Transfer Service for S3 to Cloud Storage.
- B. Manually upload the media files to Cloud Storage, run a custom script to transfer Redshift data to BigQuery, and use gsutil for transferring S3 data to Cloud Storage.
- C. Use Storage Transfer Service for both the media files and the image files, and export the Redshift data as CSV to manually import into BigQuery.
- D. Use gsutil for media files, BigQuery's built-in tools for Redshift migration, and manually upload the S3 files to Cloud Storage.



# Multiple Services

You are preparing to migrate your on-premises and cloud-based data to Google Cloud. Your data includes:

- 200 TB of media files currently stored in a network-attached storage (NAS) system.
- A data warehouse hosted on Amazon Redshift containing analytics data.
- 20 GB of image files currently stored in an Amazon S3 bucket.

You need to transfer the media files to a Cloud Storage bucket, migrate the data warehouse to BigQuery, and move the image files to a separate Cloud Storage bucket. You want to use Google-recommended tools and avoid custom coding for the migration. What should you do?

- A. Use Transfer Appliance to move the media files, BigQuery Data Transfer Service for Redshift migration, and Storage Transfer Service for S3 to Cloud Storage.
- B. Manually upload the media files to Cloud Storage, run a custom script to transfer Redshift data to BigQuery, and use gsutil for transferring S3 data to Cloud Storage.
- C. Use Storage Transfer Service for both the media files and the image files, and export the Redshift data as CSV to manually import into BigQuery.
- D. Use gsutil for media files, BigQuery's built-in tools for Redshift migration, and manually upload the S3 files to Cloud Storage.



# Multiple Services

Your company is developing a real-time feedback system for a live event, where participants will submit feedback at unpredictable rates. The system must handle varying levels of incoming feedback efficiently and process the data as soon as it arrives. You want to ensure scalability while minimizing the infrastructure management needed to process the feedback.

What should you do?

- A. Store the feedback directly in a Cloud SQL database and use Cloud Run to process the data.
- B. Send data to Pub/Sub and process data using Cloud Functions triggers to handle the feedback as it arrives.
- C. Use Compute Engine to process the feedback data in real-time and store it in Firestore.
- D. Batch the feedback data and process it periodically using Dataflow, storing the results in BigQuery.



# Multiple Services

Your company is developing a real-time feedback system for a live event, where participants will submit feedback at unpredictable rates. The system must handle varying levels of incoming feedback efficiently and process the data as soon as it arrives. You want to ensure scalability while minimizing the infrastructure management needed to process the feedback.

What should you do?

- A. Store the feedback directly in a Cloud SQL database and use Cloud Run to process the data.
- B. Send data to Pub/Sub and process data using Cloud Functions triggers to handle the feedback as it arrives.**
- C. Use Compute Engine to process the feedback data in real-time and store it in Firestore.
- D. Batch the feedback data and process it periodically using Dataflow, storing the results in BigQuery.



# Multiple Services

You are developing a real-time analytics platform on Google Cloud for processing large volumes of data generated by IoT sensors. The system continuously streams both structured and unstructured trade data from multiple global sources. You need to design a scalable and resilient architecture to handle these data streams and store them efficiently for analysis, following Google-recommended practices.

What should you do?

- A. Stream data directly into BigQuery for real-time analytics.
- B. Use Compute Engine VMs to process the data and store it in Cloud SQL.
- C. Stream data to Pub/Sub, and use Dataflow to send data to Cloud Storage.
- D. Ingest the data into Firestore for both structured and unstructured data storage.



# Multiple Services

You are developing a real-time analytics platform on Google Cloud for processing large volumes of data generated by IoT sensors. The system continuously streams both structured and unstructured trade data from multiple global sources. You need to design a scalable and resilient architecture to handle these data streams and store them efficiently for analysis, following Google-recommended practices.

What should you do?

- A. Stream data directly into BigQuery for real-time analytics.
- B. Use Compute Engine VMs to process the data and store it in Cloud SQL.
- C. Stream data to Pub/Sub, and use Dataflow to send data to Cloud Storage.**
- D. Ingest the data into Firestore for both structured and unstructured data storage.



# Multiple Services

You are preparing to migrate your company's on-premises infrastructure to Google Cloud. Your current setup includes:

- A Microsoft SQL Server cluster for your core database.
- RabbitMQ for handling messaging and event-driven workloads.
- An Oracle database for your business analytics and reporting needs.

You want to follow Google Cloud's recommended solutions to ensure global scalability, while minimizing operational overhead and infrastructure management. What should you do?

- A. Migrate the SQL Server cluster to Compute Engine, set up RabbitMQ on a managed instance, and use Cloud SQL for Oracle for the analytics workload.
- B. Move the core database to Cloud Spanner, replace RabbitMQ with Pub/Sub for event handling, and use BigQuery for analytics and reporting.
- C. Migrate the SQL Server cluster to Cloud SQL, keep RabbitMQ on Compute Engine VMs, and move the Oracle database to Bigtable for analytics.
- D. Use Cloud SQL for SQL Server, configure Kafka on GKE, and migrate Oracle to Firestore for analytical data storage.



# Multiple Services

You are preparing to migrate your company's on-premises infrastructure to Google Cloud. Your current setup includes:

- A Microsoft SQL Server cluster for your core database.
- RabbitMQ for handling messaging and event-driven workloads.
- An Oracle database for your business analytics and reporting needs.

You want to follow Google Cloud's recommended solutions to ensure global scalability, while minimizing operational overhead and infrastructure management. What should you do?

A. Migrate the SQL Server cluster to Compute Engine, set up RabbitMQ on a managed instance, and use Cloud SQL for Oracle for the analytics workload.

**B. Move the core database to Cloud Spanner, replace RabbitMQ with Pub/Sub for event handling, and use BigQuery for analytics and reporting.**

C. Migrate the SQL Server cluster to Cloud SQL, keep RabbitMQ on Compute Engine VMs, and move the Oracle database to Bigtable for analytics.

D. Use Cloud SQL for SQL Server, configure Kafka on GKE, and migrate Oracle to Firestore for analytical data storage.

